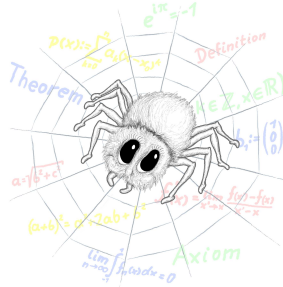


– mArachna –

Eine semantische Analyse der mathematischen Sprache
für ein computergestütztes Information Retrieval System



vorgelegt von:

Dipl.-Phys. Nicole Natho
Berlin

Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
DOKTOR DER NATURWISSENSCHAFTEN
– DR. RER. NAT. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Christian Thomsen
Berichter/Gutachter: Prof. Dr. Ruedi Seiler
Berichter/Gutachter: Prof. Dr. Manfred Stede
zusätzliche Gutachterin: Prof. Dr. Christiane Fellbaum

Tag der wissenschaftlichen Aussprache:

17. Februar 2005

Berlin 2005

D 83



Danksagungen

An dieser Stelle möchte ich mich zu allererst bei meinem Doktorvater Ruedi Seiler bedanken, der immer an meine Ideen glaubte und mir viel Freiraum verschaffte diese zu realisieren. Er unterstützte mich tatkräftig und hat keine Mühen gescheut mir weiter zu helfen.

Des Weiteren möchte ich mich bei Christiane Fellbaum bedanken, die mir durch ihre Begeisterung viel Mut machte dieses Projekt zu bearbeiten und mir viele wertvolle Ratschläge gab. Bei Sabina Jeschke, die mir ebenfalls im Zuge des Mumienprojektes Vertrauen, Zeit und wertvolle Tipps gab, meine Ideen zu verwirklichen. Bei Sebastian Rittau, der viele Ideen mitverwirklichte und diese technisch umsetzte. Bei Sven Grottke, der zahlreiche Stunden mit mir ausharrte, um dieses Werk zu vollenden. Bei Erhard Zorn, der all diese Seiten Korrektur lesen musste und immer beruhigende Worte parat hatte. Und schließlich bei Thomas Richter, der dieser Arbeit den letzten Schliff gab. Sowie an alle, die während dieser Zeit meine Launen tapfer ertragen mussten.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Motivation	9
1.2	Ziele und Hypothesen	12
2	Disziplinen	15
2.1	Einleitung	15
2.2	Wissensmanagement	18
2.2.1	Daten, Informationen und Wissen	18
2.2.2	Wissensorganisation und Wissensrepräsentation	22
2.2.3	Wissensbasierte Systeme	30
2.3	Sprachverstehen	36
2.3.1	Psychologische Modelle zum Textverstehen	42
2.4	Computerlinguistik	44
2.4.1	Morphologie	45
2.4.2	Syntax	50
2.4.3	Semantik	59
2.4.4	Fachsprache	66
2.5	Information Retrieval	69
2.5.1	Methoden des Information Retrieval	73

3	Mathematische Strukturen	77
3.1	Einleitung	77
3.2	Grundlagen	80
3.2.1	Einleitung	80
3.2.2	Mathematische Logiken	82
3.2.3	Axiomatische Mengenlehre	89
3.3	Sprachstrukturen	96
3.3.1	Entitätenebene	97
3.3.2	Binnenstrukturebene	103
3.3.3	Satzstrukturebene	107
3.3.4	Wortebene	127
3.3.5	Symbolebene	129
3.3.6	Zusammenfassung	130
3.4	Wissensstrukturen	133
3.4.1	Ontologie der Mathematik	133
3.4.2	Taxonomie der Mathematik	139
3.4.3	Zusammenfassung	149
4	Architektur von mArachna	151
4.1	Gesamtkonzeption	151
4.2	Natürlichsprachliche Analyse	155
4.2.1	Zerlegung von \LaTeX in Satzteile, Wörter und Formeln	155
4.2.2	Morphologische Analyse	162
4.2.3	Syntaktische Analyse	168
4.2.4	Semantische Analyse	176
4.3	Wissensbasis	181

<i>INHALTSVERZEICHNIS</i>	<i>7</i>
4.3.1 Grundkonzept	181
4.3.2 Realisierung	182
4.4 Information Retrieval	185
4.4.1 Grundkonzept	185
4.4.2 Realisierungsstand	186
5 Ergebnisse	189
5.1 Kritische Analyse	189
5.2 Zusammenfassung	196
5.3 Ausblick	198
6 Anhang	201
6.1 TEI-Ausgabe	201
6.2 Zerlegung der Textstruktur	204
6.3 Morphologische Analyse	208
6.4 Syntaktische Analyse	214
6.5 Semantische Analyse	218

Kapitel 1

Einleitung

Ein hinreichend klarer mathematischer Text kann in einer konventionellen Sprache formuliert werden, die nur aus einer kleinen Anzahl unveränderlicher Wörter besteht; diese Wörter werden gemäß einer Syntax kombiniert, die ihrerseits nur eine kleine Anzahl unverletzlicher Regeln umfasst; ein so formulierter Text heißt dann formalisiert.

(Nicolas Bourbaki¹)

1.1 Motivation

Nicht nur durch das Internet, sondern auch durch wissenschaftliche Publikationen wächst in der Mathematik die Menge der verfügbaren Informationen ständig an. Um auf dem neuesten Stand der Forschung zu bleiben, müssen Wissenschaftler und andere Interessenten viel Zeit für Recherchen verwenden. Leider liegt ein Großteil der vorhandenen Informationen in Form von natürlich-sprachlichen mathematischen Texten vor, so dass Computer sie nicht unmittelbar verarbeiten können. Es bietet sich daher an, den Inhalt von mathematischen Texten computergestützt zu analysieren, um relevante Informationen daraus zu

¹[Bou02, S. 164]

gewinnen. Solche Informationsextraktionssysteme existieren in verschiedensten Formen.

Im Projekt *Mumie* [SJZ04, Jes04] — Multimediale Mathematikausbildung für Ingenieure — wurde eine Lernplattform zum Einsatz von modernen Technologien entwickelt, die mathematisches Wissen in vielfältiger Weise Studenten im Präsenzunterricht zur Verfügung stellt. In diesem Zusammenhang entstanden zahlreiche Unterprojekte, darunter auch das *mArachna*-Projekt, das Gegenstand dieser Arbeit ist. *mArachna* hat sich zum Ziel gesetzt, ein *innovatives Information Retrieval System* in Form eines mathematischen Lexikons zu entwickeln. Dazu sollen natürlichsprachliche mathematische Texte maschinell verarbeitet und in ein ontologisches Modell der Mathematik eingeordnet werden. Der Anwender erhält dadurch die Möglichkeit, Anfragen zu mathematischen Konzepten und Sachverhalten zu stellen. Für die Ausgabe der Suchergebnisse sind Wissensnetze denkbar, die eine kontextuelle Einordnung der einzelnen Suchergebnisse ermöglichen. Wissensnetze sind dabei graphische Darstellungen zur Visualisierung von Begriffen und Sachverhalten und ihren Beziehungen untereinander. Im Gegensatz zu herkömmlichen Information Retrieval Systemen ist es also möglich, nach Konzepten und Ideen anstatt z. B. nach in Texten auftretenden Wörtern zu suchen. Der Schwerpunkt dieser Arbeit wird dabei nicht auf dem eigentlichen Retrieval-Interface liegen, sondern auf den grundlegenden Mechanismen zur Verarbeitung der ursprünglichen mathematischen Texte und ihrer Einordnung in eine Wissensbasis.

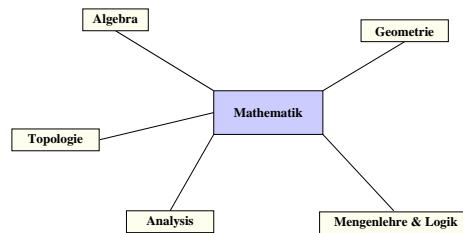


Abbildung 1.1: Wissensnetz

Zur manuellen Erstellung einer solchen mathematischen Lexikons hätten Autoren ihre Texte bereits mit den zur Vernetzung notwendigen Informationen versehen müssen. Dabei haben Autoren oft eine sehr unterschiedliche Auffassung,

wie mathematische Begriffe und Sachverhalte vernetzt werden könnten. Dies ist nicht verwunderlich, da bei einem solch komplexen Fachgebiet wie der Mathematik der Prozess der Metadatenerstellung nicht einfach zu schematisieren ist.

Das *Mumie*-Projekt speichert mathematische Informationen in Form von natürlichsprachlichen Texten in einer Datenbank. Die Grundidee der vorliegenden Arbeit ist es, diese vorhandenen Informationen zu verwenden und computerlinguistisch zu analysieren, um aus den Ergebnissen dieser Analyse ein Lexikon zu erstellen. Daher werden keine Metadaten in den betrachteten Dokumenten benötigt. Bei einer solchen inhaltsbezogenen Analyse von Texten stellt sich die Frage, wie die gewünschten Informationen effizient extrahiert werden können. Hierfür ist die mathematische Sprache gut geeignet, da sie in ihrer Struktur typische Merkmale aufweist, die eine computergestützte Analyse vereinfachen könnten. Die Sprache der Mathematik ist dabei weit mehr als nur eine Fachsprache mit charakteristischen Fachtermini. Sie ist ein künstliches Produkt des Menschen, um mathematische Sachverhalte mit klaren und einfachen Sätzen und exakt definierten Bezeichnungen darzustellen.

Die linguistische Analyse von mathematischen Texten liefert semantische Informationen in einer maschinenlesbaren Form. Zwischen semantischen Informationen und dem Aufbau von Wissensnetzen liegt ein weiter Weg. Jedoch besitzt die Mathematik glücklicherweise eine gut durchdachte, klar strukturierte Theorie, die auf der Prädikatenlogik und der Mengenlehre aufbaut. Diese Theorie besteht aus einem großflächigen Beziehungsnetzwerk, in dem die mathematischen Inhalte angeordnet werden. Dadurch entsteht eine durchgängige Strukturierung, die formalisiert werden kann. Daraus lässt sich ein Klassifikationsmodell ableiten: die Taxonomie der Mathematik. Daher sollte es möglich sein, zumindest für Teilgebiete mathematisches Wissen in Repräsentationsschemata (Wissensbasen) so zu organisieren, dass es auch durch ein Computersystem verarbeitet werden kann.

Die Repräsentationsschemata werden dabei aus komplexen semantischen Netzen bestehen. Durch diesen Aufbau ergibt sich die Möglichkeit, ein Repräsentationsmodell zu konstruieren, das mathematische Inhalte selbstständig anordnet. Voraussichtlich wird dabei keine vollständig automatisierte Analyse der Daten möglich sein, jedoch soll diese Aufgabe in wesentlichen Teilen ohne manuelle

Unterstützung durch den Computer erfolgen.

1.2 Ziele und Hypothesen

Das Grobziel von *mArachna* ist der Aufbau eines intelligenten halbautomatischen Information Retrieval Systems. Dazu sind Daten für das Information Retrieval erforderlich, aus denen die syntaktischen und semantischen Strukturen der mathematischen deutschen Sprache erkannt, analysiert und schematisiert werden können. Dazu notwendig ist die Konzeption eines Wissensrepräsentationsschemas (Ontologie), das mathematische Begriffe und Sachverhalte umfassend beschreiben und in einer Wissensbasis organisieren kann. Außerdem muss ein Konzept entwickelt werden, wie Informationen wieder aus dieser Wissensbasis extrahiert und dem Anwender in sinnvoller Art und Weise zur Verfügung gestellt werden können.

Die vorangehenden Betrachtungen legen nahe, für die *Wissensextraktion* und die *Wissensorganisation* von folgenden Hypothesen auszugehen²:

Hypothese 1.2.1

Eine computerlinguistische Analyse der mathematischen Sprache ist einfacher als die Analyse alltagssprachlicher Texte.

Diese Hypothese folgt aus der Überlegung, dass mathematische Sprache in Texten normalerweise stark strukturiert ist und dabei immer wiederkehrende Phrasenstrukturen verwendet. In diesem Zusammenhang folgt auch gleichzeitig die nächste Hypothese:

Hypothese 1.2.2

Fachsprachen wie die Mathematik besitzen wenige Ambiguitäten.

Die Mathematik verwendet wenige Begriffe aus der Alltagssprache. Neue Begriffe werden meist formal eingeführt, d. h. sie bauen auf bereits vorhandenen Begriffen und Sachverhalten auf. Dies legt die folgende Hypothese nahe:

Hypothese 1.2.3

Der Wortschatz der Mathematik ist kleiner als bei anderen Fachsprachen.

²Einige der Hypothesen wären durchaus empirisch testbar. Dies ist jedoch nicht Gegenstand dieser Arbeit.

Da die mathematische Sprache stark strukturiert ist und auf einer einfachen Logik beruht, ergeben sich für Autoren nur wenige stilistische Freiheiten:

Hypothese 1.2.4

Stilistische Elemente spielen keine entscheidende Rolle bei der linguistischen Analyse. Aufgrund des strengen Aufbaus der Mathematik ist anzunehmen, dass die stilistischen Unterschiede zwischen verschiedenen Autoren gering sind.

Seit Anfang des zwanzigsten Jahrhunderts sind Mathematiker bestrebt, die Mathematik zu axiomatisieren und zu formalisieren. Dieser Ansatz erscheint auch sinnvoll für eine Realisierung der genannten Ziele auf Computersystemen.

Hypothese 1.2.5

Das mathematische Grundlagenwissen, bestehend aus Prädikatenlogik und Mengenlehre genügt, um eine mathematische Wissensbasis aufzubauen, wie sie für ein halbautomatisches Information Retrieval System benötigt wird.

Aufgrund der vielfältigen Beziehungen zu anderen Fachdisziplinen sollen in dieser Arbeit zuerst diejenigen Disziplinen betrachtet werden, die zur Behandlung des Problems notwendig sind (Kapitel 2). Insbesondere soll dabei auf die Grundlagen der Mathematik eingegangen werden (Kapitel 3.2.2, 3.2.3). Kapitel 3.3 (Sprachstrukturen) und Kapitel 3.4 (Wissensstrukturen) sind die zentralen Kapitel. Hier werden die Strukturen zum Aufbau eines innovativen Information Retrieval Systems festgelegt. In einem letzten Schritt sollen dann anhand eines Prototypen die Ergebnisse diskutiert werden.

Kapitel 2

Disziplinen

Das einzige Mittel gegen Aberglaube ist Wissenschaft.
(Henry Thomas Buckle, engl. Philosoph, 1821-1862)

2.1 Einleitung

Zweifellos gehört die *Sprache* zu den herausragendsten Eigenschaften menschlicher Kognition. Sie ist das wichtigste Medium, um Wissen zu kommunizieren oder im Gedächtnis zu speichern. Im Hinblick auf die zunehmende Bedeutung des Computers ist es nicht verwunderlich, dass in den letzten Jahrzehnten zahlreiche Forschungsgruppen versucht haben, dem Computer das Lesen zu lehren. Leider treten dabei zahlreiche Probleme auf, die u. a. durch die verschiedensten Formen von sprachlichen Mehrdeutigkeiten (*Ambiguitäten*) induziert werden. *Fachsprachen*, insbesondere solche in denen Mathematik eine zentrale Stellung einnimmt, besitzen weniger Ambiguitäten (Kapitel 1.2, Hypothese 1.2.5). Der mathematische *Sprachstil* ist darüber hinaus im Gegensatz zum geschriebenen Sprachstil in Tageszeitungen, Belletristik, Gedichten usw. deutlich strukturierter und hierarchischer. Er besitzt einen deduktiven logischen Aufbau und besteht damit aus überschaubaren Satzstrukturen mit einfachen Aussagesätzen. Durch die vereinfachten Strukturen wird eine reduzierte Grammatik induziert, die sich leichter analysieren lässt (Kapitel 1.2, Hypothese 1.2.1). Allerdings gibt es individuelle Unterschiede zwischen verschiedenen Autoren mathematischer

Texte (*Individualstile*), die die informationstechnische Verarbeitung erschweren [Bau99].

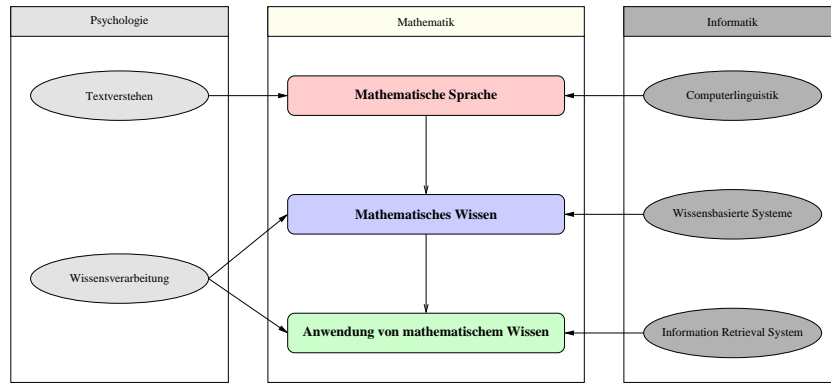


Abbildung 2.1: Überblick über die notwendigen Disziplinen

Eine computergestützte Analyse der geschriebenen mathematischen Sprache erfordert trotz der genannten Vorteile einen großen Aufwand. Verschiedene Disziplinen sind notwendig, um ein solches Konzept zu realisieren. Das theoretische Fundament bildet die *kognitive Psychologie*, die u. a. die Extraktion von Wissen aus Texten, die Speicherung von solchem Wissen und die Anwendung des gespeicherten Wissens durch den Menschen betrachtet (Kapitel 2.2, Kapitel 2.3). Dabei werden nicht biologische und chemische Betrachtungen herangezogen, sondern theoretische Modelle zur *Wissensakquisition*, *Wissensorganisation* und *Wissensnutzung* entwickelt. Diese Modelle werden als Diskussionsgrundlage verwendet, wie Wissen informationstechnisch verarbeitet werden kann. Insbesondere ist das Sprachverstehen (Kapitel 2.3) für die Computerlinguistik von Interesse.

In einen weiteren Schritt sollen kurz Methoden und Probleme der *Computerlinguistik* (Kapitel 2.4) dargestellt werden, die für die maschinelle Analyse der mathematischen Sprache von Bedeutung sind. Darüber hinaus ist die informationstechnische Wissensnutzung, die Wissen in geeigneter Weise dem Nutzer eines Computersystems zur Verfügung stellt, ein wichtiger Diskussionsgegenstand. Die technische Realisierung wird durch das *Information Retrieval* (Kapitel 2.5) gegeben, das interessante Konzepte und Analysemethoden bereitstellt.

Aus der Diskussion der Disziplinen sollte es möglich sein, ein System zu entwickeln, das mathematische Sprache analysiert und das extrahierte Wissen organisiert.

2.2 Wissensmanagement

*Zu wissen, was man weiß, und zu wissen,
was man tut, das ist Wissen.*
(Konfuzius, chin. Philosoph, 551 - 479 v. Chr.)

Wissen ist die Ressource der modernen Informationsgesellschaft, die zunehmend an Bedeutung gewinnt. Dabei ist Wissen ein wenig fassbarer Wert (*intellectual capital*) [Vä02], das in Archiven und Datenbanken in Form einer unüberschaubaren Flut von Informationen zumeist schlummert oder gar schwer zugänglich ist. Schlimmstenfalls besitzt nur ein einzelner Mensch das Wissen. Daher ist es sinnvoll, Wissen in geeigneter Form für verschiedene Personen zur Verfügung zu stellen. Heutzutage beschäftigt sich u. a. das *Wissensmanagement* (*Knowledge Management*) mit dem Begriff Wissen und seiner Verarbeitung. Dabei werden nicht fundamental neue Techniken generiert, sondern es werden bekannte Methoden aus den verschiedensten Disziplinen (Philosophie, Psychologie, Informatik usw.) verwendet.

Definition 2.2.1 (Wissensmanagement)

„**Wissensmanagement** [engl. *knowledge management*], in der modernen Organisationsführung die Gesamtheit der Modelle und Konzepte, mit denen sich die Bedeutung von Wissen als Ressource herausarbeiten sowie Techniken und Instrumente zur bewussten Gestaltung wissensrelevanter Prozesse in Organisationen entwickeln lassen. [...] Innerhalb der Datenverarbeitung bezeichnet der Begriff Wissensmanagement den Umgang mit großen, unstrukturierten Datenmengen und die Extraktion des darin enthaltenen Wissens.“ [Bro03, S. 979]

2.2.1 Daten, Informationen und Wissen

Die Basis des Wissens sind **Daten**, d. h. quantitativ aufgezeichnete Wahrnehmungen der Realität [Vä02]. Beispiele für Daten sind Texte, Bilder, Töne usw. In der Informatik sind Daten Zeichen, die mit einer Syntax versehen sind. Daten werden zu **Informationen**, indem sie in einen Kontext gestellt und zum Erreichen eines konkreten Ziels verwendet werden [Hau02]. Daher sind Informationen Daten mit einer Semantik. Allerdings wird dadurch nicht geklärt, wie Informationen Wissen formen. Um Wissen im Computer zu repräsentieren, ist

es notwendig zu definieren, was unter Wissen verstanden werden kann. Wie so häufig ist auch der Wissensbegriff nicht eindeutig definierbar. Geprägt durch Platon (im Werk *Theaitetos*) ist in der Philosophie der Begriff Wissen mit dem Wahrheitsbegriff (Kapitel 3.2.2) verbunden. In der Psychologie entspricht Wissen einer Menge von Kenntnissen (Erfahrungen), die eine Person aus ihrem Gedächtnis wiedergeben kann [And01]. Allgemein kann Wissen als gespeicherte Information betrachtet werden, aus der Schlüsse gezogen werden können, die selbst wieder als Information zum Wissen beitragen. Dabei spielt der Kontext der Information eine besondere Rolle.

Wissen := Informationen im Kontext

Um Informationen als Wissen zu organisieren, erscheint es notwendig, verschiedene Arten von Wissen zu unterscheiden, die das breite Spektrum des Wissensbegriffs eingrenzen, um somit eine übersichtliche Darstellung zu erzeugen. Nach Anderson [And01] wird in der kognitiven Psychologie menschliches Wissen in zwei Arten unterteilt:

- **Deklaratives Wissen** (Faktenwissen):
Terminologisches Wissen und Wissen über Sachverhalte; leicht verbalisierbares Wissen
- **Prozedurales Wissen** (Verarbeitungswissen):
Wissen über die Art und Weise von verschiedenen kognitiven Handlungen; schwer verbalisierbar.

Aus der Philosophie stammt ein Modell von Polanyi (Klassifikationssystem von Polanyi) [Pol85], das zwischen zwei Arten von Wissen in Bezug auf den möglichen subjektiven Charakter der Wissensaufnahme unterscheidet:

- **Explizites Wissen** (*disembodied knowledge*):
Formalisierbares und digitalisierbares Wissen
- **Implizites Wissen** (*embodied knowledge*):
Personifiziertes Wissen, das durch Erfahrung erworben wird

Ein weiteres Modell stammt von Ryle [Ryl69] und Baumgartner [Bau93] und hat ebenfalls einen philosophischen Ursprung. Es wird zwischen drei Arten von

Wissen unterscheiden, wobei die dritte Art stärker die Interpretationsfähigkeit von vorhandenen Informationen hervorhebt:

- **Faktenwissen:** deklaratives Wissen;
- **Anwendungswissen:** prozedurales Wissen;
- **Handlungswissen:**
Fertigkeiten, die sich in ausführbaren Tätigkeiten als praktisches Wissen äußern.

Alle drei Beschreibungsmodelle von Wissenstypen weisen Ähnlichkeiten auf. Sie unterteilen in leicht und schwer formalisierbares Wissen. Insbesondere wird durch das Modell von Polanyi deutlich, dass die Wissensakquisition und die Wissensnutzung einen subjektiven Charakter besitzen.

Auch in der künstlichen Intelligenz wird der Wissensbegriff ausführlich behandelt. Im Zuge der Entwicklung wurden Modelle erstellt, die versuchen menschliches Wissen im Computer zu repräsentieren. In diesem Zusammenhang befasste sich Newell [New81] u. a. mit folgenden Fragenstellungen [GRS00, S. 7]:

- „Wie kann Wissen charakterisiert werden?“
- „Wie steht eine solche Charakterisierung in Beziehung zur Repräsentation?“
- „Was genau zeichnet ein System aus, wenn es über Wissen verfügt?“

Newell kam zu der Auffassung eines dreistufigen Ebenenmodells, bestehend aus *Programmierebene*, *Symbolebene* und *Wissensebene*. Die Programmiererebene ist die unterste Ebene und wird durch die Hardware realisiert. Die Symbolebene trägt die Repräsentationen, die als Datenstrukturen und Prozesse existieren und den Wissensbestand auf der Wissensebene realisieren [GRS00, S. 7]. Der zentrale Ansatz von Newell ist die Verwendung von Logiken als fundamentalem Werkzeug (*Mathematisierung des Wissens*). Logiken werden einerseits für die

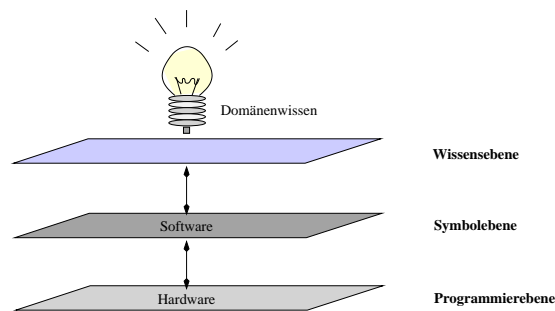


Abbildung 2.2: Newells dreistufige Ebenenmodell

Kodierung von vorhandenem Wissen verwendet, andererseits für die Implementierung von Inferenzmechanismen, um menschliche Schlussfolgerungsprozesse zu imitieren.

Die Kodierung von Wissen unter Verwendung einer logikbasierten Sprache (*Wissensrepräsentationssprache*) wird in der künstlichen Intelligenz als *Wissensbasis* bezeichnet. Die technischen Realisierungen sind jedoch immer an die Art des untersuchten Problems gebunden. Eine allgemeinere Art, Wissen darzustellen, sind *Ontologien*, die ein formales allgemeingültiges Beschreibungsmodell für bestimmte Weltausschnitte bereitstellen. Dies ist eine Beschreibung von Konzepten und ihren Beziehungen untereinander, die für eine Gruppe von Personen begriffsbildend sind.

„An ontology is a formal, explicit specification of a shared conceptualization. A conceptualization refers to an abstract model of some phenomenon. Explicit means that the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable.“ [Fen04, S. 7]

Wichtig ist der Aspekt des Informationsaustauschs. Einerseits muss eine Ontologie eine Semantik von Informationen besitzen, die maschinenverständlich ist. Andererseits muss diese Ontologie auch von Menschen akzeptiert werden. Jedoch gibt es dabei Probleme. Selbst der Mensch hat in vielen Bereichen des Lebens keine eindeutige Beschreibung für Konzepte. So ist z. B. der Begriff der „Religion“, nicht eindeutig definierbar und somit ein einheitliches Modell nicht möglich.

Konzepte beschreiben eine Menge von Objekten aus dem betrachteten Weltausschnitt, die gemeinsame Eigenschaften aufweisen. Die Instanzen sind spezielle Ausprägungen eines Konzeptes. So ist ein rotes Auto eine Instanz des Konzeptes Auto. Relationen stellen die Objekte des Weltausschnitts in Zusammenhang zueinander, z. B. durch eine „*ist_eine*“-Relation. Eigenschaften beschreiben die benutzten Konzepte und Instanzen genauer. Die Nebenbedingungen können Eigenschaften weiter einschränken z. B. in der Länge [Hel00].

Im weiteren Verlauf werden Modelle vorgestellt, die menschliches Wissen schematisieren. Häufig werden dabei assoziative Netzwerke verwendet. Wissen erscheint in diesen Modellen als Vernetzung von Informationen, die es dem Träger ermöglichen, Handlungsvermögen aufzubauen und Aktionen in Gang zu bringen. Fragen an die kognitive Psychologie betreffen dabei die Vernetzungen von Informationen und ihre Konstruktionsmöglichkeiten.

2.2.2 Wissensorganisation und Wissensrepräsentation

Wahrnehmungen aus der Umwelt werden in ein mentales Modell umgewandelt. Dabei handelt es sich um Prozesse der Klassifizierung und der Interpretation der wahrgenommenen Inhalte. Um diese internen mentalen Modelle zu beschreiben, werden in der Psychologie Konzepte entwickelt, die als *Wissensrepräsentationen* bezeichnet werden. Diese Form der menschlichen Informationsverarbeitung wird bei Anderson [And01] durch den jeweiligen Typ der Information charakterisiert. Es gibt die *bedeutungsbezogene Wissensrepräsentation*, die sich in zwei weitere Repräsentationsformen unterteilt, die *propositionale Wissensrepräsentation* und die *konzeptuelle Wissensrepräsentation*. Hierbei werden Informationen im Kontext des vorhandenen Wissens aufgenommen (Interpretation der aufgenommenen Informationen). Außerdem gibt es die *wahrnehmungsbasierte Wissensrepräsentation*, bei der Informationen vom System direkt wahrgenommen werden, beispielsweise visuell oder verbal. Diese Form der Wissensrepräsentation ist im Gegensatz zur bedeutungsbezogenen Wissensrepräsentation gut erforscht. Im Hinblick auf die Betrachtung von mathematischem Wissen soll die bedeutungsbezogene Wissensrepräsentation verwendet werden.

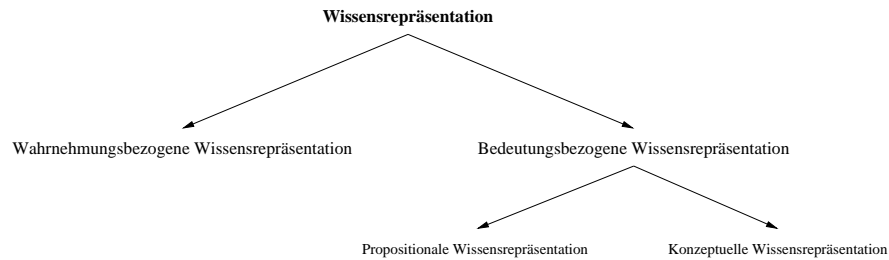


Abbildung 2.3: Wissensrepräsentationen in der kognitiven Psychologie

2.2.2.1 Propositionale Wissensrepräsentationen

Die propositionale Wissensrepräsentation ist in der kognitiven Psychologie zu einem nützlichen Modell bei der Beschreibung der Informationsverarbeitung insbesondere für natürlichsprachliche Sätze geworden. Der aus der Logik und der Linguistik übernommene Begriff *Proposition* nimmt eine zentrale Rolle ein. Eine Proposition ist die kleinste Wissenseinheit, die sich sinnvoll als wahr oder falsch beurteilen lässt. Sie entspricht somit einer Aussage. Es gibt unterschiedliche propositionale Notationssysteme, wie z. B. die *Prädikat-Argument-Struktur* von Kintsch [Kin74].

Prädikat-Argument-Struktur. Die *Argumente* entsprechen Personen, Gegenständen, Eigenschaften usw., die meistens durch Nomina beschrieben werden. Die *Prädikate* entsprechen den Beziehungen der Argumente untereinander. Sie werden vor allem durch Verben, Adjektive oder andere relationale Ausdrücke gebildet. Dargestellt wird eine propositionale Wissensrepräsentation durch eine Liste, bestehend aus einem Prädikat und den zugehörigen Argumenten.

Beispiel 2.2.1

Eine Äquivalenzrelation ist reflexiv, transitiv und symmetrisch.

Propositionale Wissensrepräsentation:

- (*ist_reflexiv*, Äquivalenzrelation)
- (*ist_transitiv*, Äquivalenzrelation)
- (*ist_symmetrisch*, Äquivalenzrelation)

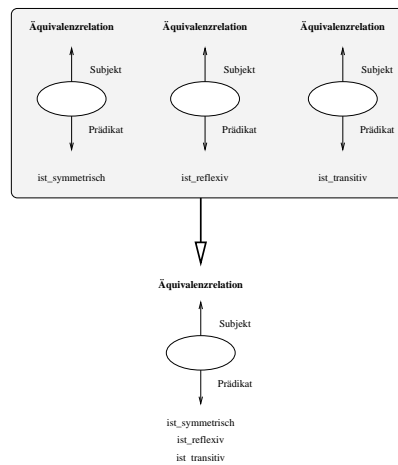


Abbildung 2.4: Graphische Realisierung eines propositionalen Netzwerks

Netzwerke aus Propositionen (*propositionales Netz*) beschreiben die Beziehungen des betrachteten Sachverhaltes. Prädikate und Argumente werden in diesem Netzwerk als *Knoten* und Pfeile als *Verbindungen* bezeichnet. Die räumliche Anordnung spielt keine Rolle. Sie können aber in hierarchischer Beziehung zueinander stehen, wobei eine Proposition als eine Einheit innerhalb einer anderen Proposition auftritt. Allerdings können sie keine allgemeinen Zusammenhänge beschreiben, die nicht explizit in dem Sachverhalt genannt werden. Experimente über die psychische Realität der propositionalen Einheiten werden in [BF71] aufgeführt.

2.2.2.2 Konzeptuelle Wissensrepräsentationen

Menschen neigen dazu, Dinge, die uns umgeben, zu ordnen, zu klassifizieren oder zu kategorisieren. Dabei werden allgemeine Merkmale einer gewonnenen Erfahrung in Kategorien, Konzepte bzw. Begriffe zusammengefasst (Abstraktion der Umwelt). Dies entspricht einer Art Mustererkennung, ohne die unsere Informationsverarbeitung überfordert wäre. Diese Mustererkennung beinhaltet Prozesse wie *Differenzierung* und *Generalisierung*. Durch die Differenzierung wird die Vielfalt der wahrgenommenen Informationen in handliche Einheiten zerlegt. Die Generalisierung ordnet diese in überschaubare Kategorien an. Somit ist die Kombination beider ein effektives kognitives Instrument.

Die einzelnen Kategorien werden basierend auf der Erfahrung des jeweiligen Menschen erstellt. Es ist egal, ob diese der Wahrheit entsprechen oder welchen Umfang sie besitzen. Kategorien sind auch nicht starr, sondern veränderbar. Dies führt in der künstlichen Intelligenz zu Problemen, da dynamische Kategorien schwer zu realisieren sind. Psychologen beschreiben Kategorien mit Attributen und Verknüpfungsregeln [Bou66]. So wird die Kategorie „*Wasser*“ mit den Attributen „*farblos*“, „*geruchlos*“, „*flüssig*“ usw. belegt. Attribute entsprechen damit relevanten Merkmalen, die einen Gegenstand charakterisieren. Verknüpfungsregeln entsprechen logischen Regeln, um die Kategorien zu strukturieren („*Wenn ..., dann*“, „*besitzt*“ usw.).

Eine wichtige Eigenschaft von Kategorien ist die Möglichkeit der Angabe einer beliebigen Anzahl von Attributen. Darüber hinaus werden viele solcher Kategorien durch den Menschen nicht eindeutig definiert. Um diese Uneindeutigkeiten zu modellieren, werden Ideale (repräsentativste Beispiele) konstruiert, die Variationen in der Interpretation zulassen (*Prototypen*) [BF71].

Zwischen den einzelnen Kategorien gibt es verschiedene Formen von Beziehungen. Um globale Zusammenhänge zu erfassen, werden Informationen in größeren kategorialen Einheiten organisiert. In konzeptuellen Wissensrepräsentationen werden Methoden vorgestellt, wie Kategorien in Beziehung zueinander stehen und wie diese im Einzelnen strukturiert sind. Hierbei wird u. a. zwischen *semantischen Netzen* ([Qui68]) und *Schemata* [BT81] unterschieden. [Wes94]

Semantische Netze. Das menschliche Gedächtnis zeichnet sich u. a. dadurch aus, dass es eine große Anzahl von Verbindungen oder Assoziationen zwischen Informationen bilden kann. Semantische Netze können solche Assoziationen beschreiben. Nach Quillian [Qui68, CQ69] speichert der Mensch Informationen über verschiedene Kategorien in hierarchischen Netzwerkstrukturen z. B. mittels *is-a*- oder *instance-of-Beziehungen*. Diese Beziehungen vermitteln die Semantik. Eigenschaften, die für Kategorien einmal angelegt werden, werden auf die darunterliegenden Hierarchieebenen vererbt. Kategorien entsprechen in diesen Netzwerken *Knoten*, und die Beziehungen entsprechen den *Kanten*.

Informationen, die nicht direkt als Kategorien gespeichert werden, müssen geschlussfolgert werden. Dazu gibt es Untersuchungen [And01, S. 155], wie sich die Abrufzeiten von geschlussfolgerten Informationen verhalten. Unter anderem

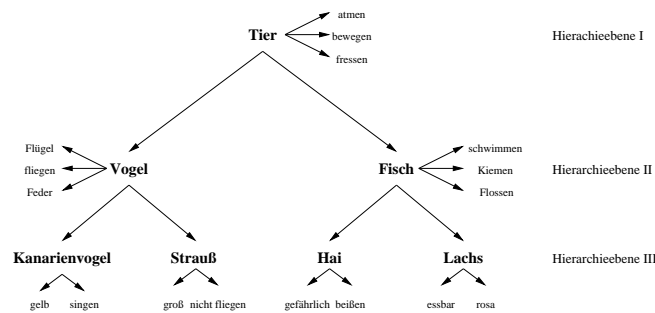


Abbildung 2.5: Graphische Realisierung einer semantischen Repräsentation nach Quillian¹ [CQ69]

benötigt der Mensch mehr Zeit, Informationen abzurufen, die nicht explizit als Kategorie gespeichert werden.

Als geeignetes Darstellungsmittel eines einfachen semantischen Netzes bieten sich *Graphen* an. Graphen sind abstrakte Strukturen, die relationale Beziehungen oder Netzwerke modellieren.

Definition 2.2.2 (Graph)

Ein **Graph** ist ein Paar $G = (V, E)$ disjunkter Mengen mit $E \subseteq [V]^2$; die Elemente von E sind also 2-elementige Teilmengen von V . Die Elemente von V nennt man die **Ecken (Knoten)** des Graphen G , die Elemente von E seine **Kanten**. Wir sagen, $G = (V, E)$ sei ein Graph auf V [Die00, S. 4].

Ein **gerichteter Graph** ist ein Paar (V, E) disjunkter Mengen (von Ecken und Kanten) zusammen mit zwei Funktionen $\text{init} : E \rightarrow V$ und $\text{ter} : E \rightarrow V$, die jeder Kante e eine Anfangsecke $\text{init}(e)$ und eine Endecke $\text{ter}(e)$ zuordnen; die Kante von e heißt dann von $\text{init}(e)$ nach $\text{ter}(e)$ **gerichtet**. Ein gerichteter Graph kann zwischen zwei Ecken x, y mehrere Kanten haben, solche Kanten nennt man **Mehrfachkanten**. Ist $\text{init}(e) = \text{ter}(e)$, so ist e eine **Schlinge** [Die00, S. 26].

Definition 2.2.3 (Weg, Zyklus)

Ein **Weg** ist ein nicht leerer Graph $P = (V, E)$ der Form

$$V = \{x_0, x_1, \dots, x_k\} \quad E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\},$$

¹Der Graph in der Abbildung kann auch Ausnahmen darstellen. So wird Vögeln die Eigenschaft *fliegen* zugeschrieben. Allerdings können z. B. Pinguine und Strauße nicht fliegen. Dies wird in dieser Graphik berücksichtigt.

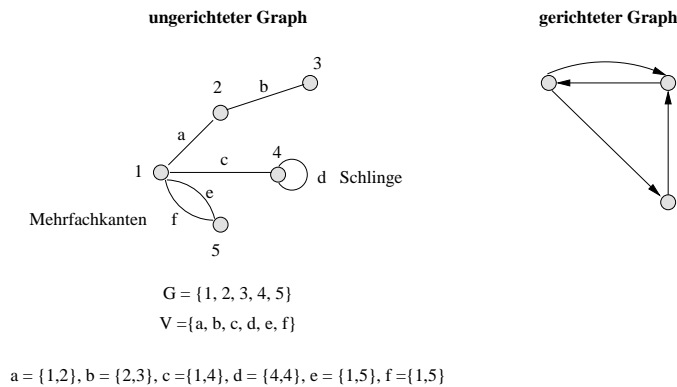


Abbildung 2.6: Darstellung verschiedener Graphen

wobei die x_i paarweise verschieden sind. Die Ecken x_0 und x_k sind die *Endecken* von P ; sie sind durch P verbunden. Die Ecken x_1, \dots, x_{k-1} sind die *inneren Ecken* von P . Die Anzahl der Kanten eines Weges ist seine **Länge** [Die00, S. 7]. Ist $P = x_0 \dots x_{k-1}$ ein Weg und $k \geq 3$, so ist der Graph $C := P + x_{k-1}x_0$ ein **Kreis** [Die00, S. 13]. Ein Graph, der keine Kreise enthält, heißt **kreislos** (**azyklisch**).

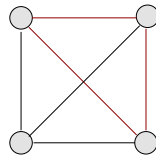


Abbildung 2.7: Darstellung der Wege eines ungerichteten Graphen

Definition 2.2.4 (Semantischen Netze)

Ein semantisches Netz (K, σ, T, τ) ist ein endlicher, gerichteter azyklischer Graph, bestehend aus:

- einer Menge K von Kategorien (Knoten des Graphen),
- einer Relation $\sigma \subseteq K \times K$ (Kanten des Graphen),
- eine Menge T von Kantentypen (mögliche Relationen zwischen den Kategorien) und

- einer Funktion $\tau : \sigma \longrightarrow T$, die jeder Kante einen Typ zuordnet.

[Rei89]

Knoten und Kanten können nicht nur Namen tragen, sondern auch Eigenschaften (Konfigurationen) haben. Komplexe Netze werden durch Hypergraphen beschrieben, d.h. durch Graphen mit Knoten, die selbst wieder Graphen sind. Durch Partitionierung und Quantifizierung kann die Ausdruckskraft von semantischen Netzen gesteigert werden.

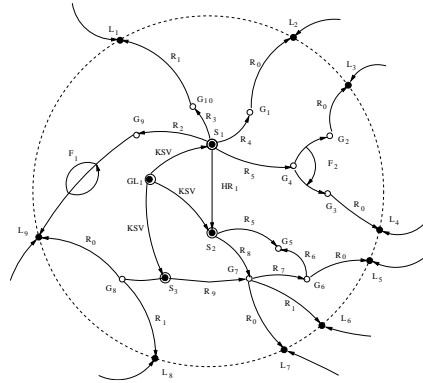


Abbildung 2.8: Bild eines komplexen Netzwerks [HB01, S. 101]

Werden semantische Netze zur Modellierung unseres Wissen verwendet, so stoßen diese bald an die Grenzen des Möglichen. Dies macht es notwendig, sie durch neue Methoden zu erweitern.

Schemata. Semantische Netze, die nur Eigenschaften von Konzepten abspeichern können, sind nicht in der Lage, die Komplexität des menschlichen Wissens zu erfassen. In *Schemata* (z.B. Haus) wird kategoriales Wissen in Form von strukturierenden Elementen (*Slots*) (z.B. Material) und deren Werten dargestellt. Die Slots definieren typische Eigenschaften (z.B. Beton), die zur Beschreibung eines Begriffs verwendet werden. Die Werte entsprechen konkreten Ausprägungen dieser Eigenschaften. Dabei besitzen die Eigenschaften üblicherweise eine oder mehrere erwartete Belegungen, die standardmäßig mit dem Schema assoziiert werden. Die Erwartungswerte können dabei zwischen verschiede-

nen Personen variieren, z. B. wird ein Stadtbewohner ein Haus eher aus Beton annehmen, während ein Bauer zuerst an Stein denkt.

Schema: Haus	
<i>Slot</i>	<i>Slotwerte</i>
Oberbegriffe	<i>Gebäude</i>
Teile	<i>Zimmer</i>
Material	<i>Stein, Holz</i>
Funktion	<i>wohnen</i>

Abbildung 2.9: Darstellung eines Schema

Der *Oberbegriffslot* in der Abbildung 2.9 ist eine spezielle Form des Slots. Er entspricht der „*is-a*“-Relation im semantischen Netzwerk und kennzeichnet die oberste Objektklasse.

Nicht nur Gegenstände weisen eine konzeptuelle Struktur auf. Wir verfügen auch über Konzepte für verschiedene Ereignisarten, z. B. „*ins Kino gehen*“. Solche Kategorien können auch mit einer Variante von Schemata dargestellt werden. Abelson et al. [AS77] entwickelten diese Variante der Schemata, die auch als *Skripts* bezeichnet werden.

In der künstlichen Intelligenz wird der *Frame*-Begriff [Min74] verwendet, er entspricht dem Begriff der Schemata, allerdings wird hierbei stärker das assoziative Modell verwendet.

Definition 2.2.5 (Frames)

Ein *Frame* ist ein Tripel (N, SN, ST) bestehend aus

- dem *Frame*-Namen N
- einer Menge nicht-terminaler Slots $SN = \{sn_1, \dots, sn_k\}$, $k \geq 1$ und
- einer Menge terminaler Slots $ST = \{st_1, \dots, st_m\}$, $m \geq 1$.

[Rei89]

Dabei sind *nicht-terminale Slots* SN die Slots, die wiederum Frames besitzen, um sich zu beschreiben. So kann z. B. in Abbildung 2.9 der Slot *Gebäude* durch

ein weiteres Frame beschrieben werden. Die terminalen Slots ST weisen dagegen als Einträge Zeichenketten auf und können nicht durch weitere Frames dargestellt werden.

Obwohl semantische Netze und Schemata ihre Vorzüge aufweisen, gelten sie nach der vorherrschende Forschungsmeinung als inadäquat, jedoch sind sie für den Fokus dieser Arbeit ausreichend.

2.2.3 Wissensbasierte Systeme

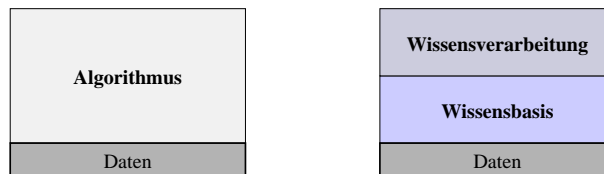


Abbildung 2.10: Unterschied zwischen Programm und wissensbasierten System

Ein künstliches System, das Wissen speichert und verarbeitet wird als **wissensbasiertes System** bezeichnet. Gewöhnliche Computerprogramme speichern implizit „Wissen“ in Algorithmen. Wird „Wissen“ verändert, so muss der Algorithmus geändert werden. Bei wissensbasierten Systemen wird dagegen streng zwischen anwendungsspezifischem Wissen (Wissensbasis, Wissensrepräsentation) und der Wissensverarbeitung (Problemlösungsstrategien, Inferenzmaschine) unterschieden. So muss bei Änderung von „Wissen“ nicht die Wissensverarbeitungskomponente geändert werden (Abbildung 2.10).

Neben der Wissensverarbeitung und der Wissensbasis besteht ein wissensbasiertes System aus weiteren Komponenten. Um Wissen in einer Wissensbasis zu speichern gibt es eine *Wissensakquisitionskomponente* (Knowledge Engineering). Als Schnittstelle für Nutzer eines wissensbasierten Systems existiert eine *Dialogkomponente* [BKI00]. Der Begriff Dialogkomponente wird häufig im Zusammenhang mit Expertensystemen genannt, daher erscheint es sinnvoll, diese Komponente zu verallgemeinern und sie als *Benutzerschnittstelle* zu bezeichnen (Abbildung 2.11).

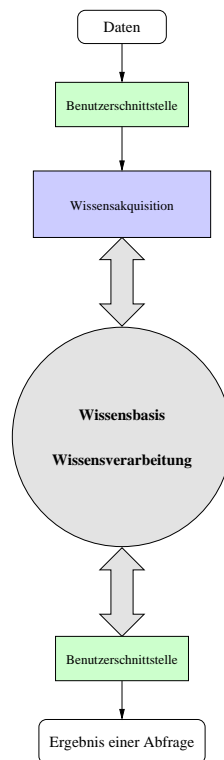


Abbildung 2.11: Aufbau eines wissensbasierten Systems

Wissensbasis. Eine Wissensbasis enthält Wissensrepräsentationen, die Wissensinhalte computergerecht darstellen. Aufgrund der unüberschaubaren Menge an Informationen ist es nützlich, eine solche Darstellung auf Teilbereiche (**Domänen**, *Weltausschnitte*) zu beschränken und nur diese Domänen zu modellieren.

Wissensbasen enthalten explizites Wissen. Explizites Wissen lässt sich auf zwei verschiedene Darstellungsarten im Computer realisieren: *deklarative* und *prozedurale Darstellung*. Deklarative Darstellungen entsprechen objektorientierten Repräsentationsformen und logischen formalen Sprachen. So können die in der kognitiven Psychologie (Kapitel 2.2) verwendeten Wissensrepräsentationsmodelle — semantische Netze, Frames und Propositionen — deklarative Darstellungen modellieren und somit auf Computersysteme übertragen werden. Prozedurale Darstellungen stellen Wissen mittels Prozeduren bzw. Anweisungen dar

und werden durch regelbasierte System realisiert. Dabei wird Wissen mit Hilfe einer Menge von Regeln (Produktionsregeln) dargestellt. Ein bekanntes Beispiel sind die Phrasenstrukturregeln der generativen Grammatik (Kapitel 2.4.2).

Wissensverarbeitung. Die Wissensverarbeitungskomponente verkörpert den Schlussfolgerungs- und Problemlösungsapparat. Sie wird auch als **Inferenzsystem** bezeichnet. Die Vorstellung der Trennung von Wissensbasis und -verarbeitung ist dabei nicht wirklich korrekt. Tatsächlich ist ein Inferenzsystem sehr wohl abhängig von dem Wissensrepräsentationsmechanismus in der Wissensbasis. Je nach Art der Repräsentation (z. B. semantische Netze, Produktionsregeln) müssen entsprechende Inferenzsysteme konstruiert werden. So muss ein Inferenzsystem auf einem semantischen Netz Suchalgorithmen auf Graphen verarbeiten können, bei regelbasierten Systemen werden dagegen die Regeln mittels Prädikatenlogik verarbeitet.

Das Inferenzsystem erlaubt es, aus vorhandenem Wissen in der Wissensbasis neues Wissen abzuleiten. Dabei werden menschliche Schlussfolgerungsprozesse imitiert. Die bekanntesten und einfachsten Systeme sind die *regelbasierten Inferenzsysteme*. Sie bestehen aus einem System von Inferenzregeln und einem Inferenzschema. Das Inferenzschema enthält die Verarbeitungsvorschriften, wie Daten aus der Wissensbasis auf die Regeln angewendet werden.

Diese Inferenzsysteme arbeiten mit der Prädikatenlogik, was gleichzeitig auch ihre Schwäche ist. Damit können keine *nicht-monotonen Schlussfolgerungen* gezogen werden, die häufig auftreten. Hierbei können Schlussfolgerungen im Laufe der Verarbeitung zurückgenommen werden. Defaultlogiken versuchen dieses Phänomen zu formalisieren. Es werden sogenannte Defaultregeln angewendet, die die eigentlichen Bedingungen über ein Objekt speichern (z. B. Vögel fliegen). Dazu kommen Ausnahmen, die durch Implikationen dargestellt werden (z. B. Pinguine können nicht fliegen).

Ein weiteres, komplexeres System für die Verwaltung von nicht-monotonen Schlussfolgerungen ist das *Truth Maintenance System* [BKI00]. Dieses System arbeitet mit *Constraints* und wird häufig als effizientes Wartungssystem für Wissensbasen verwendet.

Bei der Darstellung von unsicherem Wissen werden wahrscheinlichkeitstheoretische Methoden verwendet. Unsicheres Wissen beruht darauf, dass nicht im-

mer eindeutige Schlussfolgerungen aus Voraussetzungen gezogen werden können. Meistens gibt es keine eindeutigen Zusammenhänge. In der Sprache treten sie meist als Wörter wie *manchmal*, *beinahe* usw. auf. Es wird dem System in der Regel Wissen in Form von Unabhängigkeitsannahmen hinzugefügt, die durch Graphen repräsentiert werden. Insbesondere verwendet man daher *Markow-* und *Bayessche Netze*, aber auch Methoden der Fuzzy-Theorie [BKI00].

Kategoriebasierte Wissensbasen. Es gibt viele verschiedene Ausprägungen von semantischen Netzen. Allen gemeinsam ist, dass sie mittels Objekten, Objektkategorien und Relationen Objekte repräsentieren. Inferenzsysteme werden bei semantischen Netzen durch Vererbungsmechanismen induziert. Diese scheinbare Einfachheit kann zu komplexen Phänomenen führen, sobald Mehrfachvererbung zugelassen wird. Ein weiterer Inferenzmechanismus ist die inverse Verknüpfung, die Vererbungsmechanismen gegebenenfalls umdrehen kann. Dies bildet jedoch nur einen sehr primitiven Apparat für Schlussfolgerungen. Es gibt semantische Netze, die sehr viel komplexer sind und zusätzlich prozedurale Mechanismen benutzen, wie z. B. partitionierte semantische Netze und existentielle Graphen. Ein wichtiger Aspekt ist die Verwendung von Defaults bei Kategorien. Für solche Konstruktionen werden Beschreibungslogiken zur Beschreibung von Definitionen und Eigenschaften von Kategorien verwendet. Die wichtigste Inferenzaufgabe für Beschreibungslogiken sind:

1. die Subsumption: Überprüfung, ob eine Kategorie eine Untermenge einer anderen ist, indem ihre Definitionen verglichen werden;
2. die Klassifizierung: Überprüfungen, ob ein Objekt zu einer Kategorie gehört;
3. die Konsistenzprüfung: Überprüfung, ob die Zugehörigkeitskriterien logisch erfüllbar sind.

Allerdings fehlen den Beschreibungslogiken die Mechanismen der Negation und Disjunktion.

Wissensakquisition. In irgendeiner Form müssen Informationen dem System zugeführt und organisiert werden. Die Wissensakquisitionskomponente bietet dafür eine Schnittstelle an. Sie kann einerseits ein Eingabeinstrumentarium

für Experten sein, andererseits auch eine (semi-) automatische Anwendung, welche aus Datenanalysen geeignete Einträge erzeugt. Ein wichtiger Prozess ist auch die fortlaufende Wartung der Wissensbasis.

Die Wissensakquisition beinhaltet Prozesse der Wissenserhebung, die relevantes Wissen in Texten erkennt, der Wissensinterpretation, die das erhobene Wissen interpretiert, und der Wissensformalisierung, die das Wissen computergerecht aufbereitet.

Benutzerschnittstellen. Die Benutzerstellen bieten verschiedenste Arten von Kommunikation zwischen einem Nutzer und dem System an. Einerseits gibt es Instrumentarien für die Wissensakquisition, andererseits geht es um die Ausgaben der Wissensbasis aufgrund von Anfragen an das System. Dabei kann es sich um graphische Ausgabe, menügesteuerte Dialoge, formalsprachliche oder natürlichsprachliche Mittel handeln. Diese Benutzerschnittstelle kann auch einen Teil eines Information Retrieval Systems darstellen (Kapitel 2.5).

Ontologien. Eine neuere Entwicklung ist die Verwendung von Ontologien bei der Erstellung von wissensbasierten Systemen. Ontologien stellen eine Grundlage für viele innovative wissensbasierte Systeme bereit. Ein Vorteil bei der Verwendung einer Ontologie besteht in der gemeinsamen und effizienten Kommunikationsbasis zwischen Nutzer und System und zwischen unterschiedlichen Systemen.

Ontologiesprachen, die Ontologien beschreiben, verwenden Regeln auf Konzepten, Eigenschaften von Konzepten und Relationen zwischen Konzepten sowie zusätzliche Sprachmittel. Es gibt eine ganze Reihe solcher Ontologiesprachen (einfach, framebasiert, logikbasiert). Sie werden oftmals graphisch dargestellt.

Eine der bekanntesten Ontologiesprachen ist die *Ressource Description Framework* (RDF) [RDF]. Eine weitere Sprache ist DAML+OIL, die sich aus den beiden Sprachen DAML (*DARPA Agent Markup Language*) [DAR03] und OIL (*Ontology Inference Layer*) [Ont] entwickelt hat. Sie baut auf dem RDF Schema auf, bietet aber größere Ausdrucksmöglichkeiten.

Eine im Rahmen des semantischen Webs entstandene Ontologiesprache ist OWL (*Web Ontology Language*) [OWL]. Auch diese Sprache hat das RDF Schema zur

Grundlage. Sie ist eine durch das W3C standardisierte Sprache mit fest definierter Syntax und Semantik. Sie basiert auf der Beschreibungslogik (*description logic*) und auf Konzepten von DAML+OIL.

2.3 Sprachverstehen

Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt. Die Logik erfüllt die Welt; die Grenzen der Welt sind auch ihre Grenzen. [...] Was wir nicht denken können, das können wir nicht denken; wir können also auch nicht sagen, was wir nicht denken können.

(L. Wittgenstein, österr. Philosoph, 1889 - 1951)

Das Sprachverstehen von geschriebenen Texten ist ein komplexer und vielschichtiger Konstruktionsprozess, der bis heute nicht vollständig verstanden ist. „Bei unseren alltäglichen Gesprächen verarbeiten und verstehen wir eine große Anzahl von Sätzen; erfolgreiches Verstehen ist eine Voraussetzung für eine effektive Kommunikation und stellt die Grundlage unserer sozialen Interaktion dar. Aus diesen Gründen ist es von besonderer Bedeutung, das Wissen und die Verarbeitungsvorgänge zu erforschen, die uns zu einem Verständnis der Sprache befähigen.“ [Wes94, S. 296]

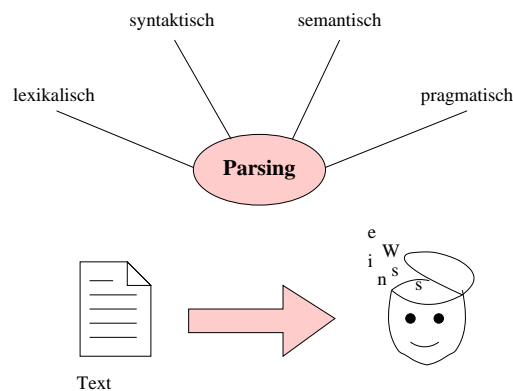


Abbildung 2.12: Überblick Textverstehen

Sprachverstehen bzw. Textverstehen bedeutet nach M. Pinkal [GRS00, S. 739] die „Gewinnung von Bedeutungsinformationen aus einer gesprochenen oder geschriebenen Eingabeäußerung“ und besteht aus einer Kombination von lexikalischen, syntaktischen, semantischen und pragmatischen Analysen. Der Prozess

der Analyse wird als **Parsing** bezeichnet, „durch den die Wörter [...] in eine mentale Repräsentation überführt werden [...]“. [And01, S. 389]

Beim Parsing treten allerdings Probleme auf, wann und wo einzelne Analyseschritte gemacht werden und wie mehrdeutige Strukturen aufgelöst werden. In den folgenden Betrachtungen sind die einzelnen Analysen nicht von der semantischen Interpretation zu trennen, da jede experimentelle Anordnung zum Sprachverstehen zwangsläufig auf die Semantik Bezug nehmen muss.

Semantische und pragmatische Analyse. Unter Textverstehen (semantische Interpretation) wird der „Vorgang der Festlegung von Bedeutungen [verstanden]. Nach dieser Definition hat [ein Mensch] einen Satz nur dann verstanden, wenn er beispielsweise den in dem Satz enthaltenen Instruktionen folgen kann oder eine angemessene Frage stellt.“ [Wes94, S. 311]. Bei der semantischen Interpretation wird zwischen **wörtlicher** und **indentierter Bedeutung** unterschieden. Die wörtliche Bedeutung bezieht sich allein auf die Aussage eines Satzes. Die indentierte Bedeutung dagegen beschreibt die Interpretation der Aussage des Satzes (Kontextinformationen). Es muss daher wörtliche und indentierte Bedeutung gleichzeitig erfasst werden, um einen Satz zu verstehen. Zweideutigkeiten (**Ambiguitäten**) treten dann auf, wenn eine der beiden Bedeutungen nicht erfasst wird.

Dennoch besitzen Menschen die Fähigkeit, aus einzelnen Wörtern die Bedeutung eines Satzes zu erkennen. So ist der Satz, „Eis Kinder mögen“, verständlich. Insbesondere verlassen sich Kinder eher auf semantische als auf syntaktische Muster [SN74]. Erwachsene dagegen integrieren semantische und syntaktische Komponenten, um Sätze zu interpretieren. Dies tun sie in einem kontinuierlichen Prozess.

Es existieren viele Wörter und Sätze, die mehrere Interpretationen zulassen, da es sich entweder um ambige Wörter oder um ambige syntaktische Konstruktionen handelt. Dabei unterscheidet man zwischen **lokalen** (vorübergehenden) und **globalen** (anhaltenden) **Ambiguitäten**. Die lokale Ambiguität bezieht sich auf Zweideutigkeiten, die sich im Satzverlauf wieder auflösen. Bei der globalen Ambiguität können frühstens am Ende des Satzes die Zweideutigkeiten wieder aufgelöst werden (**Garden-Path-Satz**). Die Garden-Path-Sätze sind ein

wichtiger Beleg, dass Menschen versuchen, einen Satz unmittelbar zu interpretieren.

Es gibt aber auch feststehende Wortgruppen (**Phrasen**), die als Ganzes interpretiert werden [PF79]. So beziehen sich z. B. Quantifizierer („weniger“, „alle“ usw.) auf das nachfolgende Substantiv und tragen somit zu der semantischen Interpretation des Substantivs bei und umgekehrt [Hö83, NC00]. Aber es gibt auch viele Wortgruppen, die eine idiomatische Bedeutung haben, die sich nicht unbedingt aus der Summierung der Einzelbedeutungen ergibt, wie z. B. „*blinder Passagier*“.

Aber nicht nur aus einzelnen Sätzen werden semantische Informationen gewonnen, sondern auch satzübergreifend. „*Die Schwierigkeit der Integration von Informationen aus verschiedenen Sätzen hängt zum Teil von der Struktur der Botschaft ab, die von dem Sprecher ausgeht.*“ [Wes94, S. 320] Das Verstehen eines Textes geschieht dann besser, wenn die einzelnen Sätze im gleichen Kontext zueinander stehen. Für die Zusammensetzung von Sätzen existieren Regeln der Kommunikation zwischen zwei Kommunikationspartnern („*given-new contract*“) [CH77]. Diesem Ansatz zufolge schließen Kommunikationspartner einen Vertrag, neue Informationen so zu übertragen, dass der Zuhörer diese leicht in sein Vorwissen integrieren kann. Dabei gibt es augenfällige syntaktische Konstruktionen, die neue Begriffe einführen oder sich aus dem Kontext ergeben. So bezieht sich z. B. ein Personalpronomen auf die genannte Person oder den Gegenstand im Satz davor. „*Das implizite Wissen dieser Regeln ist ein wichtiger Bestandteil unseres pragmatischen Wissens.*“ [Wes94, S. 321]

Lexikalische Analyse. In der lexikalischen Analyse geht es um die Identifizierung des semantischen Inhalts eines einzelnen Wortes in einem Text. Dies kann nur durch eine Kombination von morphologischen, syntaktischen und semantischen Analysen geschehen. Als Resultat werden Wissensinhalte über Wörter und ihre Bezeichnung generiert.

In Untersuchungen [Wes94] wurde festgestellt, dass die lexikalische Analyse ein nicht-trivialer Prozess ist. So können bei Wörtern wie *Fliege* Mehrdeutigkeiten im Verständnis des Wortes (**lexikalische Ambiguität**) auftreten. Es kann nicht eindeutig geklärt werden, wie Menschen diese Ambiguitäten auflösen. Untersuchungen zeigen, dass die Disambiguierung einerseits durch die Kontextinfor-

mationen und andererseits durch persönliche Präferenzen der einzelnen Personen beeinflusst wird. Welcher Einfluss in bestimmten Situationen bevorzugt wird, kann nicht geklärt werden.

Syntaktische Analyse. Der Mensch besitzt trotz seines endlichen mentalen Vermögens ein riesiges Reservoir an sprachlichem Wissen, mit dem er eine unendliche Anzahl von Sätzen bilden und verstehen kann. Noam Chomsky [Cho57, Cho67] versuchte dieses Phänomen zu ergründen und zu schematisieren. Er stellte eine endliche Anzahl von Regeln auf, mit denen korrekte sprachliche Sätze gebildet werden. Das beschriebene Regelsystem nannte er **generative Grammatik**, auch als **Phrasenstrukturgrammatik** bekannt.

Bei der Phrasenstrukturgrammatik werden Satzbausteine (*Konstituenten*) hierarchisch organisiert. Die Konstituenten spiegeln dabei die innere syntaktische Struktur eines Satzes wider und werden nach bestimmten Regeln (*Phrasenstrukturregeln*) zu einem Satz zusammengefügt. In der Abbildung 2.13 werden anhand des Satzes

„Ein Vektor ist ein Element eines linearen Raumes.“,

folgende Phrasenstrukturregeln angewendet, um den Satz zu strukturieren:

Ein Satz (S) besteht aus einer Nominalphrase (NP) und Verbalphrase (VP):

$$S ::= NP \ VP$$

Eine Nominalphrase besteht aus einem Substantiv (N), oder einem Artikel (DET) und einem Substantiv, oder aus einem Adjektiv (ADJ) und einem Substantiv, oder aus einem Artikel, einem Adjektiv und einem Substantiv.

$$NP ::= [DET] \ [ADJ] \ N$$

Eine Verbalphrase besteht aus einem Verb (V) und einer Nominalphrase.

$$VP ::= V \ NP$$

Die Analyse des obigen Satzes erfordert ebenfalls eine Rekursion der Nominalphrase. Aus einer Konstituente NP wird eine Folge abgeleitet, die dieselbe Konstituente wieder enthält.

$$\text{NP} ::= \text{NP NP}$$

Die Zerlegung lässt sich graphisch durch einen Ableitungsbaum (*Phrasenstrukturbaum*) darstellen.

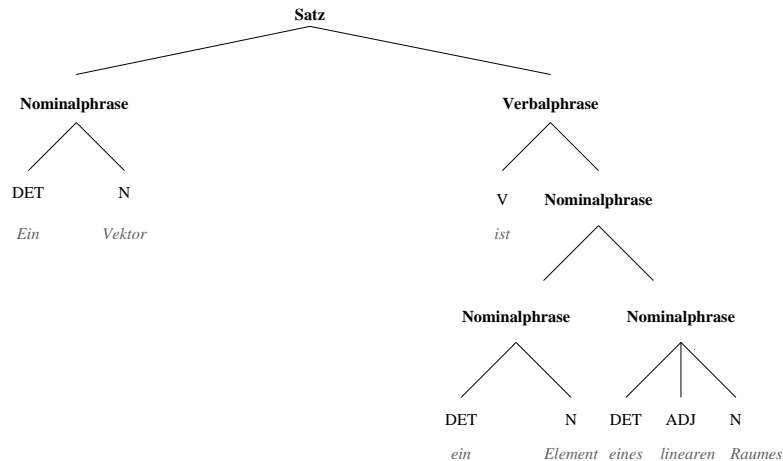


Abbildung 2.13: Phrasenstrukturbaum

$$[S [NP [_{DET} \text{Ein}] [_{N} \text{Vektor}]] [_{VP} [_{V} \text{ist}] [NP [_{DET} \text{ein}] [_{N} \text{Element}]] [NP [_{DET} \text{eines}] [_{ADJ} \text{linearen}] [_{N} \text{Raumes}]]]]]$$

Allerdings können viele *Satzarten* [Dud98] nicht allein durch die Phrasenstrukturgrammatik beschrieben werden. Dies wurde von Chomsky erkannt, und er führte daher die **Transformationsgrammatik** ein. Durch *Transformationsregeln* können in dieser Grammatik z.B. Sätze im Passiv in Sätze im Aktiv umgewandelt werden, ohne den semantischen Inhalt zu verlieren. Der Aktivsatz lässt sich dann durch die bekannten Phrasenstrukturregeln analysieren. Sätze besitzen daher in der Transformationsgrammatik mindestens zwei strukturelle Ebenen:

1. **Oberflächenstruktur:** Die Struktur eines Satzes, wie er normalerweise in Texten auftritt.

2. **Tiefenstruktur:** Abstrakte syntaktische Basis eines Satzes oder Satzbau-
steins, die alle notwendigen semantischen und syntaktischen Informationen
enthält (Konstituentenstruktur).

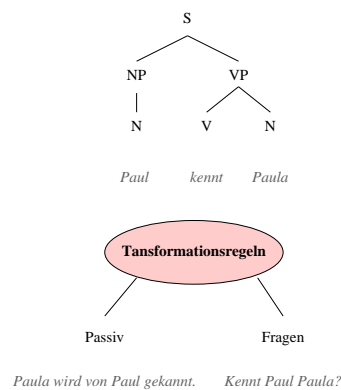


Abbildung 2.14: Transformationsregeln

Dies bedeutet für die Anwendung, dass Sätze von ihrer Oberflächenstruktur in die zugrundeliegende Tiefenstruktur transformiert werden. Es wird somit deutlich, dass die Sprache eine komplexe Struktur besitzt, die sich nicht ohne weiteres durch ein einfaches Modell beschreiben lässt. Daher muss ein mehrschichtiges Modell verwendet werden, um die Sprache adäquat darzustellen.

Auch bei syntaktischen Analysen treten Mehrdeutigkeiten (**syntaktische Ambiguität**) auf. Hierbei gibt es wiederum unterschiedliche Meinungen zur Disambiguierung [Wes94]. Die **Satzteilhypothese** besagt, dass zuerst die syntaktische Struktur eines Satzes analysiert und dann dessen semantische Bedeutung determiniert wird. In diesem Fall müssen einer lesenden Person alle möglichen syntaktischen Ambiguitäten eines Satzes bekannt sein. Im anderen Fall werden Kontextinformationen verwendet, um eine eindeutige syntaktische Struktur auszuwählen.

Eine lesende Person müsste daher alle anderen Möglichkeiten von syntaktischen Strukturen nicht bewusst wahrnehmen. Untersuchungen zeigen, dass die Satzteilhypothese sehr fragwürdig ist. Vermutlich ist die syntaktische Disambiguierung ein kontinuierlicher und fortlaufender Prozess und beeinflusst alle Teile eines Satzes und nicht nur größere syntaktische Einheiten.

2.3.1 Psychologische Modelle zum Textverstehen

Mentale Modelle. Um einen Text zu verstehen, muss der Text in ein Modell überführt werden, das in das bestehende Modell einer Person (Vorwissen, Weltmodelle) eingeordnet werden kann. Nach Johnson-Laird [JL83, JL95, JL00] existiert neben einer mentalen Repräsentation eines Textes eine nicht-sprachliche Repräsentation (**mentales Modell**). Dieses mentale Modell ist eine personifizierte Repräsentation der im Text beschriebenen Situation, die über den reinen Inhalt des Textes hinausgeht [RS99]. Somit enthält das mentale Modell Inferenzschlüsse, die durch die Interaktion zwischen der mentalen Repräsentation des Textes und dem Weltwissen der Person entstehen (Kohärenzbildung). Dadurch beschreibt das mentale Modell den Prozess des Verstehens von natürlichsprachlichen Texten.

Untersuchungen zu der Bestätigung dieses Modells können bei [GML87, GL92, OG96] gefunden werden. Allerdings werden keine genaue Angaben gemacht, wie dieses Modell im Gehirn repräsentiert wird. Sie beschreiben nur den Prozess der Inferenzbildung beim Sprachverstehen.

Textverstehen mit propositionalen Modellen. Im Gegensatz zu den mentalen Modellen wurden die propositionalen Modelle ([Kin74, KvD83, Kin88, Kin98]) direkt im Hinblick auf das Sprachverstehen entwickelt². Es wird davon ausgegangen, dass der Mensch im Verlauf des Verarbeitungsprozesses den Text in einzelne Aussagen (*Propositionen*) zerlegt. Diese Propositionen werden extrahiert und miteinander in Relation (temporale, konditionale Relationen usw.) gestellt (*Textbasis*) (Kapitel 2.2). Des Weiteren existiert eine erwartete semantische Repräsentation des Lesers gegenüber dem gelesenen Text (*Situationsmodell*) [Kin88]. Auf Grundlage dieser beiden Modelle, Textbasis und Situationsmodell, wird eine *mentale Textrepräsentation* gebildet, die den Text interpretiert. Untersuchungen zu diesem Modell können bei [KK73, WMF95] gefunden werden.

Allerdings können dadurch keine semantischen oder syntaktischen Mehrdeutigkeiten aufgelöst werden. Dazu ist immer eine semantische Analyse notwendig. Ebenfalls beschreiben propositionale Modelle nicht, dass das Textverstehen

²Sie wurden auf Grundlage der generativen Grammatik von Chomsky über den „case-grammar“-Ansatz von Fillmore (1968) entwickelt.

ein dynamischer Prozess ist, in dem fortlaufend Inhalte umgearbeitet werden müssen.

Textverstehen mit interaktiven Modellen Bei den *interaktiven Modellen* wird davon ausgegangen, dass der Leser eines Textes über bestimmte Situationen Modelle speichert. Diese Modelle beinhalten Hintergrundwissen und die Erwartungen zum Informationsgehalt des Textes. Beide Modelle stehen miteinander in Wechselwirkung. Aus dieser grundlegenden Betrachtungsweise entstanden z. B. die *Skripttheorie* von Schank und Abelson [AS77] und die *konnektionistischen Modelle* [WP85].

Die mentalen und propositionalen Modelle sind zwei entgegengesetzte Darstellungen der gängigen Modelle [Sch93, RS99]. Tatsächlich gehen die meisten Forscher davon aus, dass im Prozess der Sprachverarbeitung verschiedene Repräsentationen aufgebaut werden, die nebeneinander existieren [GMZ97, RSS02].

2.4 Computerlinguistik

Er hat mit menschlichen Zügen überrascht.

(G. Kasparow (*1963), aserbeidschanischer Schachspieler, nachdem er gegen den Schachcomputer *Deep Blue* verloren hatte, 2002)

Die *Linguistik* ist die „*Wissenschaft, die sich mit der Struktur und dem Funktionieren der Sprache befasst*“ [Bro03, S. 536]. Linguisten betrachten zwei grundlegende Konzepte der Sprache, die *Produktivität* und die *Regelhaftigkeit* [And01]. Die Produktivität beschreibt die vielfältigen Konstruktionen, mit denen aus Wörtern Sätze gebildet werden können. Die Regelhaftigkeit schränkt die Produktivität ein, indem sie nur eine endliche Anzahl von Wortkombinationen zulässt. Diese Regelhaftigkeit einer Sprache wird durch ihr Regelsystem ausgedrückt, das als *Grammatik* bezeichnet wird. Die Grammatik unterteilt sich in die Teilgebiete *Phonologie*, *Syntax*, *Semantik* und *Morphologie* (siehe Abbildung 2.15).

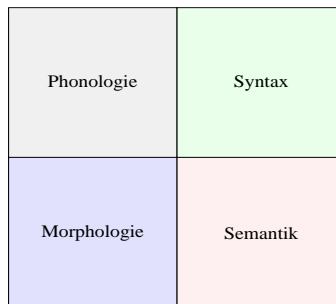


Abbildung 2.15: Bausteine der Grammatik

Die Phonologie untersucht die Funktionen der einzelnen Laute und Lautgruppen. Die Beziehungen der sprachlichen Elemente im Satz werden durch die Syntax beschrieben. Mit der Vermittlung der Bedeutung der einzelnen Textbausteine befasst sich die Semantik. Die Morphologie beschreibt die Formveränderung der Wörter durch Deklination und Konjugation.

Nach Anderson [And01] ist es ein Ziel der Linguisten, „*ein Regelsystem [Grammatik] zu erstellen, das die strukturellen Regelhaftigkeiten einer Sprache erfasst*.“ [And01, S. 356]

2.4.1 Morphologie

Die Morphologie (Formenlehre) „*untersucht die systematischen Beziehungen zwischen Wörtern und Wortformen bzw. Regeln, nach denen Wörter bzw. Wortformen gebildet werden*“ [CEE⁺01, S. 175]. Die Ergebnisse aus der morphologischen Analyse werden auch für die syntaktische Analyse verwendet, so dass die Morphologie einen wichtigen Beitrag zum gesamten Prozess des Sprachverstehens (Kapitel 2.3) liefert.

Das *Wort* (Lexem) bildet eine selbstständige lexikalische abstrakte Einheit, aus der sich verschiedene *Wortformen* ableiten lassen. Die Menge aller Wortformen wird als *Paradigma* bezeichnet. Diese Wortformen werden aus dem zu betrachtenden Wort durch folgende Transformationen gebildet [Hau00]:

1. *Flexionsmorphologie: Die Bildung von Wörtern durch systematische Variationen, durch die sich ein Wort an die verschiedenen syntaktischen Umgebungen (Numerus, Person, Kasus, Tempus, Genus, Komparation, Modus) anpasst.*

„Buch“: „Buch – es“, „Buch – e“, „B[ü]ch – er“, ...

2. *Wortbildungsprozess:*

- *Komposition: Die Bildung eines neuen Wortes durch die Zusammensetzung von Wörtern.*

„Eisen – Tür“³, „Eintag(s) – Fliege“⁴

- *Derivation: Die Bildung eines neuen Wortes auf Basis eines einzelnen Lexems mit Hilfe eines Affixes (Suffixes und/oder Präfixes).*

„Zwerg“: „Zwerg – lein“, „zwergenhaft“, „zwergenartig“, ...

Morpheme sind die kleinsten bedeutungstragenden (grammatischen) Einheiten, die in endlicher Anzahl in einer Sprache auftreten. Zu einem Morphem existieren *Allomorphe*. „*So wie Sätze genau genommen aus Wortformen (und nicht aus Wörtern) bestehen, so bestehen Wortformen genau genommen aus Allomorphen*“

³Nominalkompositionen machen 2/3 des dt. Wortschatzes aus [Dud98]

⁴Fugenlaut

(und nicht aus Morphemen)“ [Hau00, S. 271]. Die Grundformen (Wurzeln) bilden die Grundlage für die Derivation und Flexion. Die peripheren Morpheme sind die Affixe.

Beispiel 2.4.1

$\text{buch}_{\text{Morphem}} = \{\text{buch}, \text{büch}\}_{\text{Allomorphe}}$

Morphologische Analyse. Eine zentrale Aufgabe der morphologischen Analyse „besteht darin, die kleinsten sprachlichen Einheiten mit Bedeutung, die Morpheme, zu ermitteln und ihre strukturellen Eigenschaften als Bausteine von Wörtern zu beschreiben (strukturelle Linguistik)“ [fLuLdUB04].

Die strukturalistische Linguistik zerlegt nicht nur die linguistische Analyse in ihre Bestandteile — Semantik, Syntax und Morphologie — sondern stellt auch Verfahren bereit, Wörter zu zerlegen. Dadurch können relevante Anteile von Wörtern separiert und identifiziert werden. Daraus ergeben sich zwei Analyseschritte: Segmentierung und Klassifizierung [fLuLdUB04].

Aus der *generativen Morphologie*, die ebenfalls aus dem Strukturalismus hervorgeht, können weitere morphologische Analyseformen erschlossen werden. Die Beschreibungsmethoden werden dabei stark durch die syntaktische Analyse geprägt. Nach Spencer [SZ98] werden drei Ansätze unterschieden:

1. Morphembasierter Ansatz:

Kombination von Morphemen zu einem Wort

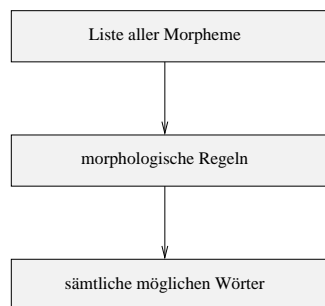


Abbildung 2.16: Morphembasierter Ansatz

2. Wortbasierter Ansatz:

Kombination von Grundmorphemen mit Affixen zu einem Wort

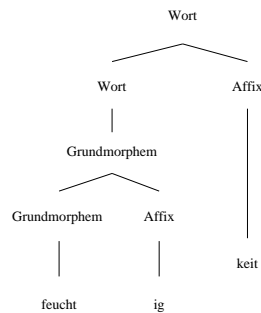


Abbildung 2.17: Wortbasierter Ansatz

3. Realisierungsbasierter Ansatz:

Kombination von Allomorphen zu einem Wort

Realisierung. Anhand der Ansätze von Spencer können verschiedene morphologische Analysen auf dem Computer realisiert werden. Hierfür bieten sich *endliche Automaten* an, die für die morphologische Analyse durch die *Finite-State Transducer* realisiert werden.

Definition 2.4.1 (Endlicher Automat)

Ein **endlicher Automat** wird durch ein 5-Tupel $M = (Z, \Sigma, \delta, S, F)$ dargestellt, wobei

1. Z : endliche, nicht-leere Menge von möglichen Zuständen,
2. Σ : endliches, nicht-leeres Eingabealphabet,
3. $\delta : Z \times \Sigma \longrightarrow Z$ (Überföhrungsfunktionen),
4. $S \in Z$: Anfangszustand,
5. $F \subseteq Z$: Menge der Endzustände

bezeichnen. [Hed02, S. 57]

Ein **Alphabet** Σ ist eine nicht-leere, endliche Menge. Elemente eines Alphabets heißen Zeichen (Symbole, Buchstaben). Die n -Tupel (x_1, \dots, x_n) von Zeichen $x_i \in \Sigma$ heißen **Wort** über dem Alphabet Σ . Die Menge aller Wörter über einem Alphabet Σ heißt Σ^* . Die leere Folge (Länge 0) wird mit ϵ bezeichnet und heißt leeres Wort. Σ^+ ist die Menge aller nicht-leeren Wörter über dem Alphabet: $\Sigma^+ := \Sigma^* \setminus \{\epsilon\}$. Die einfachste Operation auf Wörtern ist die Konkatenation (Verkettung) $\bullet : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ mit $x \bullet y = x_1, \dots, x_m, y_1, \dots, y_n$ mit $x, y \in \Sigma^*$

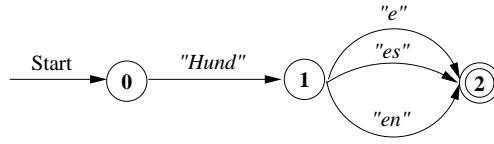


Abbildung 2.18: Darstellung eines endlichen Automaten

Das Zustandsdiagramm dieses endlichen Automaten ist ein gerichteter Graph mit

1. Knoten entsprechen den Zuständen (0, 1 und 2).
2. Jedes Paar $(q, a) \in Z \times \Sigma$, für das δ definiert ist, entspricht einer *gerichteten* Kante von q nach $q' := \delta(q, a)$ ($\delta(0, \text{"Hund"}) = 1$, $\delta(1, \text{"e"}) = 2$, $\delta(1, \text{"es"}) = 2$, $\delta(1, \text{"en"}) = 2$).
3. Anfangszustand (0).
4. Endzustand (2).

Endliche Automaten sind sehr einfache Maschinenmodelle, die z. B. ein Wort morphemweise abarbeiten. Nur wenn das Wort korrekt ist, erreicht der endliche Automat seinen Endzustand.

Ein *Finite-State Transducer* ist ein endlicher Automat, der gleichzeitig zwei Symbole (allgemein: n Symbole) bearbeitet und in entsprechende Zustände übergeht.

Definition 2.4.2 (Finite-State Transducer (FST))

Ein *Finite-State Transducer* ist ein (deterministischer) endlicher Automat $M = (Z, \Sigma, \delta, S, E)$ mit $\Sigma \subset X_1 \times X_2^5$.

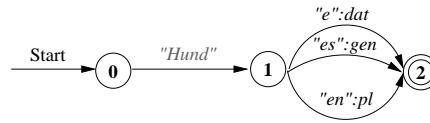


Abbildung 2.19: Darstellung eines Finite-State Transducer

Finite-State Transducer verarbeiten also geordnete Paare von Symbolen aus zwei verschiedenen Alphabeten. Somit existiert die Möglichkeit, ein Wort in verschiedene Wortformen zu transformieren. Es ist aber auch möglich, den umgekehrten Weg zu betrachten, indem man das zu untersuchende Wort in seine Bestandteile zerlegt.

Ein Modell, das die Morphologie mit Finite-State Transducern beschreibt, ist die *Zwei-Ebenen Morphologie* [Kos83]. Die Zwei-Ebenen Morphologie besteht aus zwei Ebenen der Analyse: Oberflächenebene und lexikalische Ebene. Die Oberflächenebene enthält die Wortformen, die in einem Text vorkommen. Die lexikalische Ebene enthält die morphologischen Informationen. Wie diese aufgebaut sind – Nennform, Morpheme usw. – ist nicht vorgeschrieben. Die Zwei-Ebenenregeln konstruieren Abbildungen zwischen den Ebenen, die durch Finite-State Transducer beschrieben werden. Somit besteht das System aus einem Lexikon und einem Regelsystem.

Eingabe: Oberfläche	→	Ausgabe: lexikalische Ebene
Hunde	↦	Hund + Plural + Maskulin + Nominativ

Eine weitere morphologische Analysemethode ist die vererbungsbasierte Repräsentationssprache *DATR* [EG96]. Diese speichert morphologische Informationen in einer Graphenstruktur. Knoten entsprechen Wörtern in Nennform. Der Pfad ist durch morphosyntaktische Informationen – <form, präs, sg>: 1. Person Singular Präsens – gegeben. Der Wert des Pfades entspricht dann der Wortform. Die Pfade bilden grundlegende Regeln, mit denen über Inferenzen weitere Wortformen gebildet werden können.

⁵ X_1, X_2 sind Alphonete

Beispiel 2.4.2

WATEN:

<form>	==	<wurzel> <endung>
<wurzel>	==	wat
<endung präs sg eins>	==	e
<endung präs pl zwei>	==	et

[CEE⁺01, S. 191]

Beim *Wortklassentagging* werden allen Wörtern eines Textes Tags zugeordnet. Tags enthalten morphosyntaktische Informationen. „*Beim Tagging sollen Formen im Kontext morphosyntaktisch disambiguiert, d. h. mit der im Kontext intendierten Beschreibung versehen werden.*“ [GRS00, S. 678]

Ein Methode für das Wortklassentagging ist das *regelbasierte Tagging*. Dabei wird ein Regelsystem aus einem vorgegebenen Textkorpus erstellt. Diese einfachen Regeln werden direkt auf einen Text angewendet. Der Vorteil dieser Methode ist die einfache Erweiterbarkeit des Regelsystems. Allerdings müssen diese Regeln von Hand erstellt werden. Die Betrachtung von neuen Domänen führt meistens zu einer Neubildung des Regelsystems.

Ein weiteres Verfahren ist das *statistische Tagging* (z. B. Brill-Tagging [Bri92, Bri94]). Bei diesem Lernverfahren werden aus bereits getaggten Wörtern automatisch Regeln abgeleitet.

2.4.2 Syntax

Die Morphologie (Kapitel 2.4.1) beschränkt sich auf die Analyse der einzelnen Wörter. Die Syntax nutzt die morphologische Analyse und macht Voraussagen darüber, ob die Sätze einer Sprache grammatikalisch korrekte Formen besitzen oder welche Aneinanderreihungen von Wörtern und Satzzeichen syntaktisch korrekte Sätze ergeben. Demnach beschäftigt sie sich also mit dem „*Bau von Wortgruppen und Sätzen*“ [Dud98, S. 609].

Definition 2.4.3 (Satz)

„*Sätze sind sprachliche Einheiten, die relativ selbstständig und abgeschlossen sind. Sie bauen sich aus kleineren sprachlichen Einheiten auf, die ihrerseits auch*

schon einen gewissen Selbstständigkeitsgrad haben, aus Wörtern und gegliederten Wortgruppen; und sie erscheinen normalerweise in größeren selbstständigen und abgeschlossenen sprachlichen Einheiten.“ [Dud98, S. 609]

In der Einleitung wurde besprochen, dass die Regelmäßigkeit die Anzahl der möglichen Kombinationen einschränkt und somit nur eine endliche Kombination von Wortkombinationen zu Sätzen führt. So gibt es z. B. nach Helbig [HB01, S. 445] endlich viele „morphologisch-syntaktische Satzstellungsglieder“:

1. Verbkonstruktionen: *Finites Verb („Ich gehe.“), Infinitiv des Verbs („Ich werde gehen.“), Partizip des Verbs („Ich bin gegangen.“), Präposition und Verb („Alle hielten die Regel für gelungen.“)*
2. Substantivkonstruktionen: *Nominativ des Substantivs, Akkusativ des Substantivs, Dativ des Substantivs, Genitiv des Substantivs, Präposition und Substantiv („Die Mutter wartet vor der Schule.“)*
3. Adjektivkonstruktionen: *Adjektiv, Adjektiv und Präposition („Der Professor hält das Thema für interessant.“)*
4. Adverbkonstruktionen: *Adverb („Es sitzt dort.“), Präposition und Adverb („Sie kommt von dort.“)*

Es gilt, solche syntaktischen Strukturen zu erkennen und zu schematisieren, damit sie computergerecht nutzbar gemacht werden können. Ziel der Linguisten ist es daher, Theorien zur Beschreibung von Gesetzmäßigkeiten zu finden, die die Syntax einer Sprache möglichst vollständig beschreiben können.

Syntaktische Analyse. Die syntaktische Analyse stellt methodische Prinzipien bereit, um syntaktische Strukturen menschlicher Sprache zu erfassen. Nach Carstensen [CEE⁺01] werden *syntaktische Strukturen* unter zwei Gesichtspunkten betrachtet:

- *Dependenzsyntax*
- *Konstituentenstruktursyntax*

Bei der Dependenzsyntax werden syntaktische Strukturen als Relationen zwischen Wörtern betrachtet. Das Verb nimmt dabei eine Schlüsselposition ein, da alle anderen Wörter im Satz vom ihm abhängen. Die grammatikalischen Ursachen liegen in der *Valenz* der Verben. Die Valenz beschreibt, ob Verben im Satz ergänzungslos gebraucht werden oder nicht. Je nach Art der Ergänzungen werden bestimmte Valenzklassen unterschieden (Akkusativobjekt, Dativobjekt usw.). Inhaltlich beschreiben Verben Tätigkeiten, Zustände u. ä. Dies beeinflusst die Angaben von Personen und Dingen.

Aus den beschriebenen Betrachtungen leitet sich dann die *Dependenzgrammatik* ab.

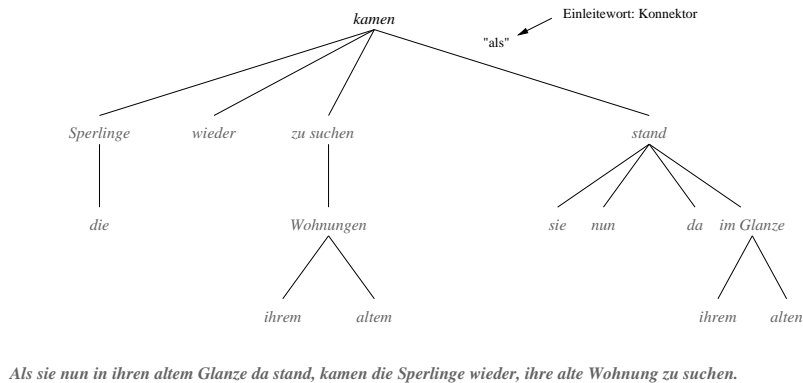


Abbildung 2.20: Dependenzgrammatik [CEE⁺01, S. 204]

Die Konstituentenstruktursyntax nimmt dagegen an, dass neben den Wörtern noch weitere komplexere Strukturen existieren. Diese komplexen Strukturen werden als *Phrasen* bzw. *Konstituenten* bezeichnet. Konstituenten lassen sich dann wieder in einzelne Wörter zerlegen.

Konstituentenstrukturen sind schon im Kapitel 2.3 in der syntaktischen Analyse betrachtet worden (generative Grammatiken) [Cho57, Cho67]. Generative Grammatiken spiegeln den hierarchischen Konstituentenaufbau wider. Jede Konstituente gehört zu einer syntaktischen Kategorie, die in einer bestimmten Reihenfolge zusammengesetzt wird. Konstituenten sind: *Nominalphrase* (NP), *Verbalphrase* (VP), *Präpositionalphrase* (PP) und *Adjektivphrase* (AP). Die zugehörigen Kategorien setzen sich zusammen aus:

Satz:	S: NP VP
Nominalphrase:	NP: (Artikel) + (Adjektiv) NP: Pronomen
Verbalphrase:	VP: Verb + NP VP: Verb + PP
Präpositionalphrase:	PP: Präposition + NP
Verb:	V: Hilfsverb + Verb

Die Abbildung der Phrasen in die Kategorien bezeichnet man als *Phrasenstrukturregel* und die Abbildung der Kategorien (Substantiv, Verb usw.) auf terminale Wörter heißt *lexikalische bzw. terminale Regel*. Probleme bei der syntaktischen Analyse treten durch strukturelle Ambiguitäten auf. So kann z. B. ein Nebensatz so eingebunden sein, dass es schwer ist, eine eindeutige Interpretation zu erhalten.

Realisierung. Für die Analyse von Sprachen ist die *Theorie der formalen Sprache* unerlässlich und somit werden Sprachen über ihre Grammatik beschrieben. Eine formale Sprache L über einem Alphabet Σ ist eine beliebige Teilmenge von Σ^* .

Definition 2.4.4 (Grammatik)

Eine **Grammatik** ist ein Quadrupel $G = (\Sigma_N, \Sigma_T, S, R)$ bestehend aus:

1. Σ_N, Σ_T : endliche, nicht-leere Mengen mit $\Sigma_N \cap \Sigma_T = \emptyset$ mit Σ_N nicht-terminale Symbole und Σ_T terminale Symbole;
2. R : endliche Menge von Erzeugungsregeln (Grammatikregeln, Produktionsregeln) der Form $\alpha \rightarrow \beta$ mit $\alpha \in (\Sigma_T \cup \Sigma_N)^+$ und $\beta \in (\Sigma_T \cup \Sigma_N)^*$;
3. einem Startsymbol $S \in \Sigma_N$.

(nach [Hed02, S. 29])

Grammatikregeln beschreiben die Struktur der Wörter einer Sprache. Ausgehend von der Satzstruktur werden durch die Regeln die Strukturen immer weiter verfeinert bis terminale Symbole entstehen. Die Zwischenstrukturen heißen dann grammatikalische Kategorien Σ_N .

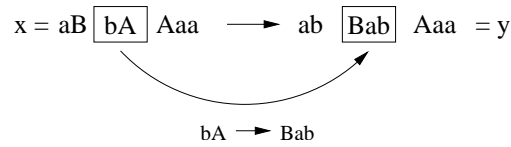


Abbildung 2.21: Beispiel aus [Hed02, S. 32]

Nach Chomsky existieren verschiedene Typen von Grammatiken (Chomsky-Hierarchie). Es werden im Rahmen dieser Arbeit nur kontextfreie Grammatiken betrachtet, die sich dadurch auszeichnen, dass sie effiziente Parsingalgorithmen ermöglichen.

Definition 2.4.5 (Kontextfreie Grammatik)

Eine Grammatik $G = (\Sigma_N, \Sigma_T, S, R)$ heißt kontextfrei (oder auch Typ-2), falls alle Grammatikregeln von der Form $A \longrightarrow \alpha$ mit $A \in \Sigma_N^+$ und $\alpha \in (\Sigma_N \cup \Sigma_T)^+$ sind. [CEE⁺01, S. 72]

Ein graphisches Beispiel wird gegeben durch den Ableitungsbaum in Abbildung 2.22.

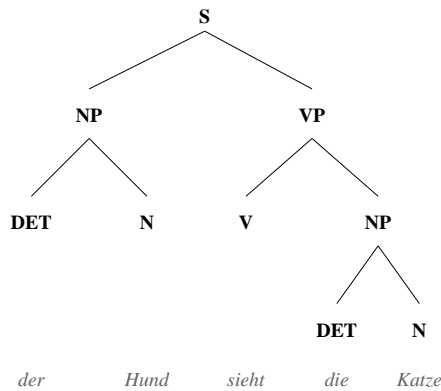


Abbildung 2.22: Ableitungsbaum

$$G = \langle \{ S, NP, VP, DET, N, V \}, \{ der, Hund, bellt, die, Katze \}, S, R \rangle$$

$$R = \{ S \rightarrow NP VP, NP \rightarrow DET N, VP \rightarrow V NP, DET \rightarrow der, \\ DET \rightarrow die, N \rightarrow Hund, N \rightarrow Katze, V \rightarrow sieht \}$$

„Die Wurzel des Baumgraphen ist mit dem Startsymbol S (für Satz) etikettiert. Der Satz besteht aus zwei Teilstrukturen, der Nominalphrase (NP) 'der Hund' und der Verbalphrase (VP) 'sieht die Katze'. Die NP 'der Hund' besteht aus dem Determinierer (DET) 'der' und dem Nomen (N) 'Hund'. Die VP besteht aus dem Verb (V) 'sieht' und der NP 'die Katze', die dieselbe interne Struktur aufweist wie die Subjekts-NP.“ [CEE⁺01, S. 207]. Allerdings werden durch die Verwendung von kontextfreien Grammatiken viele grammatikalisch inkorrekte Konstruktionen zugelassen. Deshalb werden grammatikalischen Kategorien (Phrasen) etwas erweitert bzw. differenziert hinsichtlich des Tempus, Kasus usw. (z. B. Genusinformationen: N_{maskulin} , N_{feminin} und N_{neutrum}) und somit entstehen auch neue Regeln. Das Problem ist hierbei allerdings, dass die Regelmenge exponentiell anwächst.

Ein Prototyp für kontextfreie Grammatiken ist der *ATN-Formalismus* (augmented transition network) [Woo80]. Hierbei handelt es sich um ein rekursives Transitionsnetzwerk (RTN), das aus mehreren Teilnetzen besteht. Die Teilnetze verkörpern sowohl terminale (z. B. Nomen und Adjektive) als auch nicht-terminale Anteile (z. B. NP und VP).

Der **constraintbasierte** bzw. **unifikationsbasierte Ansatz** basiert auf Merkmalsstrukturen. Dadurch werden komplexe Objekte durch mehrere Eigenschaften charakterisiert⁶.

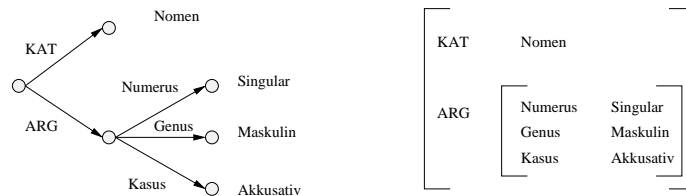


Abbildung 2.23: Merkmalsstruktur

Zu den prominentesten Vertretern dieses Ansatzes gehören die *Generalized Phrase Structure Grammar* (GPSG) [GKPS85], die *Lexical Functional Grammar* (LFG) [Bre82], *Parsing and Translation* [Shi86] sowie die *Head Driven Phrase Structure Grammar* (HPSG) [PS94].

⁶Auch Chomsky verwendet schon Merkmalsstrukturen um Verben zu subkategorisieren.

Die **GPSG-Grammatik** verwendet verschiedene Restriktions- und Regelformate. Es gibt ID-Regeln und LP-Statements. Die ID-Regeln ähneln den traditionellen Phrasenstrukturregeln, jedoch ist die Reihenfolge der Wörter nicht festgelegt. So ist sowohl $NP \rightarrow DET\ N$ als auch $NP \rightarrow N\ DET$ möglich. Erst durch die Anwendung eines LP-Statement $DET < N$ wird die Reihenfolge festgelegt. Die grammatikalischen Kategorien bestehen aus Merkmalsstrukturen. Die Werte von Merkmalen sind atomar. Hauptkategorien sind Verben, Substantive, Adjektive und Pronomen; Nebenkategorien sind Konjunktionen, Interjektionen und Gradpartikel. Die vier Hauptkategorien können den Kopf einer Phrase bilden. Bei der HPSG wird zusätzlich davon ausgegangen, dass der Kopf einer Phrase ein wichtiges Element ist. So ist der Kopf der Verbalphrase das Verb. In der HPSG gibt es dafür keine Phrasenstrukturregeln mehr, sondern allgemeine „*hierarchisch organisierte lexikalische Strukturen*“ [CEE⁺01, S. 216].

Die **lexikalisch-funktionale Grammatik** ist ein Gegenmodell zu Chomskys Standardtheorie der generativen Grammatik. Hierbei werden jeder grammatikalischen Beschreibung eines Satzes zwei Strukturen zugeordnet: die Konstituentenstruktur (*C-Struktur*) und die funktionale Struktur (*F-Struktur*). Die C-Struktur wird durch Phrasenstrukturregeln erzeugt. Diese Strukturen werden dann auf die F-Struktur abgebildet, die die grammatikalischen Funktionen (z.B. Subjekt) als Merkmale repräsentiert.

Eine wichtige Eigenschaft der LFG ist, dass grammatikalische Strukturen nicht mehr durch syntaktische Regeln, sondern durch lexikalische Regeln bestimmt werden.

Das **Parsing and Translation (PATRII)** ist der einfachste unifikationsbasierte Grammatikformalismus. PATRII besteht aus zwei Komponenten: einem Lexikon und einem Regelsystem. Die Regeln bestehen aus kontextfreien Phrasenstrukturregeln und einer Menge von Pfadgleichungen, die den kontextfreien Regeln Restriktionen auferlegen. Dabei werden die syntaktischen Kategorien durch Merkmalsstrukturen repräsentiert.

Beispiel 2.4.3 (PATRII)

Aus dem Satz „Der Hund bellt die Katze an.“ kann die Nominalphrase „der Hund“ mittels der Regel $NP \rightarrow DET\ N$ zerlegt werden. In PATRII kann diese Regel beschrieben werden durch:

$$X_0 \rightarrow X_1 X_2$$

mit den Pfadgleichungen:

$$R = \begin{cases} < X_0 CAT > = NP \\ < X_1 CAT > = DET \\ < X_2 CAT > = N \end{cases}$$

wobei CAT die Bezeichnung für eine Kategorie ist (Abbildung 2.24).

In diesem Beispiel enthalten die Argumente ARG die Genus- und die Numerus-Informationen. Es existiert die zusätzliche Regel $< X_1 ARG > = < X_2 ARG >$ (Abbildung 2.24). Durch den Vergleich der Argumente der Merkmalsstrukturen des Artikels (DET) und des Substantivs (N) wird bei Übereinstimmung der Argumente (Unifikation) angenommen, dass der Artikel zusammen mit dem Substantiv verwendet werden kann.

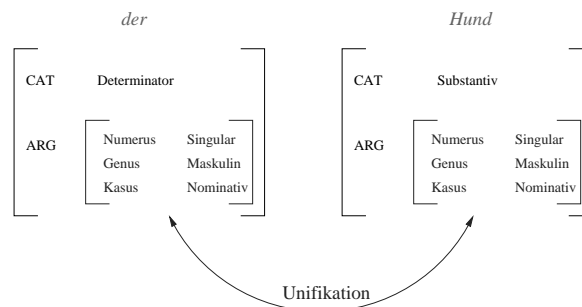


Abbildung 2.24: Beispiel Unifikation

In der **probabilistischen kontextfreien Grammatik** enthält jede Regel als zusätzliches Element eine Wahrscheinlichkeit.

Beispiel 2.4.4

Ein Satz muss immer aus einer Verbalphrase und einer Nominalphrase bestehen und somit besitzt diese Regeln den Wahrscheinlichkeitswert 1. Eine Nominalphrase kann auf verschiedene Weise zerlegt werden, die Wahrscheinlichkeit, dass die Zerlegung in Artikel und Substantiv erfolgt, ist 0.65 [CEE⁺01]

$$\begin{aligned} S &\longrightarrow VP NP \quad 1 \\ NP &\longrightarrow DET N \quad 0,65 \end{aligned}$$

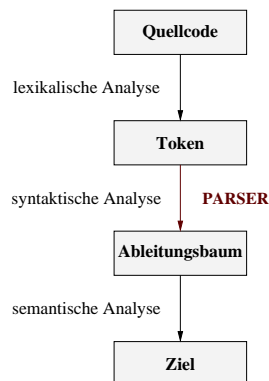


Abbildung 2.25: Parsing für eine computerlinguistische Analyse

Die Wahrscheinlichkeit einer Ableitung relativ zu einer gegebenen Kette von terminalen Symbolen wird als Produkt der Einzelwahrscheinlichkeiten aller Regeln definiert, die in der Ableitung vorkommen.

Parsing. Da im Rahmen dieser Arbeit kontextfreie Grammatiken verwendet werden, können Parsingformalismen für diese Grammatiken verwendet werden. Als Ergebnis des Parsens entsteht ein Ableitungsbaum, der dann durch die semantische Analyse den Zielcode generiert.

Bei der Konstruktion solcher Bäume können Mehrdeutigkeiten auftreten. Eine kontextfreie Grammatik G heißt *mehrdeutig*, falls es ein Wort in der durch G generierten Sprache L gibt, das mindestens zwei verschiedene Ableitungsbäume hat. Beim Aufbau eines Ableitungsbaums durch den Parser gibt es verschiedene Strategien. Nach [CEE⁺01, S. 225] existieren folgende Möglichkeiten:

1. *Verarbeitungsrichtung*: Links-Rechts-Verarbeitung, Head-Corner Parsing
2. *Analyserichtung*: Top-down-Verarbeitung, Bottom-up-Verarbeitung
3. *Suchstrategie*: Tiefensuche, Breitensuche

Die Entwicklung von Parsingalgorithmen für natürliche Sprache ist nicht einfach, da komplexe Strukturen erkannt werden müssen und außerdem Ambiguitäten auftreten können. Oft reichen dafür sequentielle Verfahren, bei denen

die einzelnen Schritte getrennt und nacheinander ausgeführt werden, nicht mehr aus, und es ist nötig, auf integrative Verfahren zurückzugreifen, in denen die Schritte nebeneinander ausgeführt werden und sich gegenseitig beeinflussen.

2.4.3 Semantik

Die Semantik ist eine grammatikalische Komponente, die sich mit der inhaltlichen Bedeutung von Sprachbausteinen beschäftigt. Eine formale Definition ist schwierig, da nicht eindeutig zu klären ist, was eigentlich die Bedeutung eines Wortes, Satzes oder Textes ist.

In den logischen Sprachen ist der Begriff Semantik eindeutig definiert. So ist die Semantik in der Aussagenlogik gegeben durch eine Funktion, die jeder Aussage die Werte wahr oder falsch zuordnet (Kapitel 3.2.2).

So ist die Aussage $x \wedge y$ wahr, wenn x und y wahr sind, aber in allen anderen Fällen falsch. Die Semantik definiert also Regeln, wie die Wahrheitswerte einer Aussage im Hinblick auf ein zu betrachtendes Modell ermittelt werden.

„Ziel einer semantischen Theorie ist eine Analyse derjenigen Prozesse, die es einem Rezipienten einer Äußerung ermöglichen, die Ideen und Gedanken hinter diesen Äußerungen zu verstehen“ [CEE⁺01, S. 246].

Zur Vereinfachung der semantischen Analyse werden verschiedene Semantikarten nach ihren Anwendungsgebieten unterschieden. Die **lexikalische Semantik** (Wortsemantik) beschäftigt sich mit dem Bedeutungsinhalt der einzelnen Wörter. Die **Satzsemantik** behandelt den Bedeutungsinhalt eines Satzes. Der Bedeutungsinhalt eines ganzen Textes wird durch die **Diskurssemantik** beschrieben.

Semantische Analyse. Die semantische Analyse ist zweifellos die wichtigste, aber auch schwierigste Analyse, die beim Textverstehen durchgeführt werden muss. Sie stützt sich bei der Ermittlung semantischer Informationen auf die vorhergehenden morphologischen und syntaktischen Analysen.

Grundlage für die Analyse des semantischen Inhalts der Sprache ist das *Kompositionalitätsprinzip* von G. Frege:

„Die Bedeutung eines komplexen Ausdrucks ist eine Funktion der Bedeutungen seiner Teile und der Art ihrer Kombination.“

[CEE⁺01, S. 248]

Durch das Kompositionalitätsprinzip besteht die Möglichkeit, die durch die syntaktische Analyse entstandenen syntaktischen Einheiten (Phrasen und Wörter) semantischen Komponenten zuzuordnen. Aus der Interpretation der einzelnen Komponenten wird eine Gesamtinterpretation des semantischen Inhaltes konstruiert.

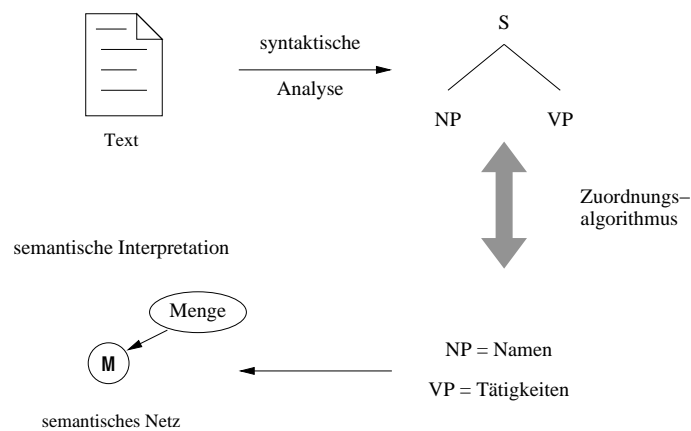


Abbildung 2.26: Grundlegendes Prinzip einer semantischen Analyse

Die **formalen Semantiken**, die auf logikbasierten Formalismen operieren, verwenden das Kompositionalitätsprinzip. Dabei wird nicht nur die Prädikatenlogik verwendet, sondern auch Typenlogik, intensionale Logik, Modallogik usw. Dadurch können auch Erklärungs- und Folgerungsbeziehungen (Inferenzmechanismen) angewendet werden. Formale Logiken werden genutzt, um natürlich-sprachliche Texte zu analysieren, da sie schematisierbar sind und somit einen informationstechnischen Zugang ermöglichen.

Nach Pinkal [GRS00, S. 740] muss eine semantische Analyse folgende elementaren Aufgaben erfüllen:

1. *Semantikkonstruktion*: Erstellung von semantischen Informationen auf Grundlage syntaktischer und morphologischer Informationen;

2. *Semantische Resolution*: Ermittlung von Disambiguierungen und Kontextinformationen aus dem Text;
3. *Semantische Auswertung*: Extraktion von relevanten semantischen Informationen mit Hilfe von Deduktions- und Inferenzmechanismen;

Realisierung. Voraussetzung für eine semantische Analyse ist ein semantischer Formalismus, der informationstechnisch realisiert werden kann. Formale Logiken erfüllen diese Voraussetzung. Es stellt sich die Frage, welche Logiken sprachliche Oberflächen semantisch beschreiben können. Die Prädikatenlogik erster Stufe bietet solche Möglichkeiten und wird daher häufig eingesetzt. Sie ist auch die einfachste Form der semantischen Analyse.

Der Beschreibungsformalismus der Prädikatenlogik besteht aus Individualsymbolen (z. B. Eigennamen) und Prädikatensymbolen (z. B. Verben, Adjektive). Prädikate entsprechen somit Eigenschaften von Objekten.

„Hansi rechnet“	=	rechnen(Hansi)
Hansi	=	Individualsymbol
rechnen	=	Prädikat

Jedes Prädikat hat eine Stelligkeit, die die Anzahl der Argumente angibt. Zum Beispiel ist **schlafen(x)** einstellig und **größer_als(x,y)** zweistellig, wobei x und y Variablen für Individualsymbole sind. Satzbedeutungen werden mittels prädikatenlogischer Formeln beschrieben. Aus aussagenlogischen Junktoren und Quantoren (Kapitel 3.2.2) können weitere Formeln gebildet werden: z. B. **schlafen(x) \wedge schlafen(y)**. Konnektoren entsprechen dabei Konjunktionen (z. B. „und“, „oder“) und Quantoren Determinatoren (z. B. „ein“, „alle“).

Beispiel 2.4.5

„alle rechnen“ $\Rightarrow \forall x$ **Rechnen(x)**

Durch $\forall x \phi$ besitzt der Ausdruck ϕ den Skopus (Wertebereich) des Quantors $\forall x$ wird somit zu einer gebundenen Variable.

Eine Domäne wird in der Logik durch ein **Modell** repräsentiert. Ein Modell der Prädikatenlogik $\mathcal{M} = (\mathcal{D}, F)$ besteht aus [CEE⁺01, S. 40]:

1. \mathcal{D} : eine nicht-leere Menge (Domäne)

2. F : eine Abbildung mit Eigenschaften:

- (a) Jedem n -stelligen Prädikat wird eine n -stellige Relation zugeordnet.
- (b) Jedem Individualsymbol wird ein Element aus \mathcal{D} zugeordnet.

„Hans rechnet.“	und	„Hansi liest ein Buch.“
$\mathcal{D} = \{\text{Ha}, \text{Hi}, \text{MB}\}$		Prädikate = $\{\mathbf{Rechnen}(\mathbf{x}), \mathbf{Lesen}(\mathbf{x}, \mathbf{y})\}$
$F(\text{Ha}) = \text{„Hans“}$		$F(\text{Rechnen}) = \{\text{Hans}\},$
$F(\text{Hi}) = \text{„Hansi“}$		$F(\text{Lesen}) = \{\text{Hansi}, \text{Mathebuch}\}$
$F(\text{MB}) = \text{„Mathebuch“}$		

Eine semantische Interpretation $[[\]]$ einer Variablen v oder eines Individualsymbols c bezüglich eines Modells \mathcal{M} und einer Funktion g weist jeder Variablen einen Wert aus \mathcal{D} zu. Die Interpretation wird formal beschrieben durch:

1. $[[v_i]]_g^{\mathcal{M}} = g(v_i)$ für alle Variablen v_i ;
2. $[[c_i]]_g^{\mathcal{M}} = F(c_i)$ für alle Konstanten c_i ;
3. Interpretationsregeln z. B. $[[\phi \wedge \psi]]_g^{\mathcal{M}} = 1$ gdw $[[\phi]] = 1$ und $[[\psi]] = 1$;
4. aussagelogische Regeln und Äquivalenzregeln (Kapitel 3.2.2).

Das Problem der prädikatenlogischen Beschreibung ist offensichtlich. Der Repräsentationsmechanismus ist nicht ausdrucksstark genug, um die vielen Phänomene der natürlichen Sprache zu beschreiben. So existiert keine eindeutige Zuweisung von Wahrheitswerten in der natürlichen Sprache. Die Variablen x und y des zweistelligen Prädikates **lesen**(\mathbf{x}, \mathbf{y}) können nicht zwischen Subjekt und Objekt des Satzes unterscheiden und die Prädikatenmodifikatoren wie z.B. „sehr *schlaues Kind*“ können nicht beschrieben werden.

Eine weitere Beschreibungsmöglichkeit wird durch die Typenlogik gegeben. Die Typenlogik besitzt einen Mechanismus, um syntaktische Strukturen bestimmten Typen von semantischen Strukturen zuzuordnen. Solche semantischen Beschreibungsmöglichkeiten werden durch die **Montague Semantik** [Mon74] beschrieben. Allerdings beschreibt sie nur den Bereich der Satzsemantik.

In der Typenlogik werden Konstanten und Prädikaten Typen so zugeordnet, dass sie sich intern unterscheiden. Dabei gibt es die Basistypen e (*entity*) und t (*truth value*). Hat ein Ausdruck A den Typ $\langle \alpha, \beta \rangle$, so bedeutet das: *A nimmt Ausdrücke vom Typ α und bildet mit ihnen zusammen Ausdrücke vom Typ β .* Dies wird durch die folgenden Beispiele verdeutlicht:

1. Einstellige Prädikate wie z. B. **Schlafen(x)** nehmen Ausdrücke vom Typ $\langle e, t \rangle$ an und bilden sie auf Wahrheitswerte ab:

$$\begin{array}{ccccc} \mathbf{Schlafen(x)} & + & \text{Peter} & = & \mathbf{Schlafen(Peter)} \\ \langle e, t \rangle & + & e & = & t \end{array}$$

2. Zweistellige Prädikate bilden Individualkonstanten auf einstellige Prädikate ab.

$$\begin{array}{ccccc} \mathbf{Lieben(x,y)} & \mapsto & \mathbf{Lieben(Peter)} & \rightarrow & \mathbf{Lieben(Peter)(Maria)} \\ \langle e, t \rangle & \mapsto & & \rightarrow & \langle e, \langle e, t \rangle \rangle \end{array}$$

Eine solche Typisierung ist nicht deterministisch, sondern zu einem Ausdruck lassen sich unter Umständen viele verschiedene Typisierungen finden. Für jeden Typ existiert eine Domäne \mathcal{D}_{typ} , wobei [CEE⁺01, S. 49]:

1. $\mathcal{D}_e = \mathcal{D}$
2. $\mathcal{D}_t = \{0, 1\}$
3. $\mathcal{D}_{\langle \alpha, \beta \rangle}$ ist die Menge aller Funktionen von \mathcal{D}_α nach \mathcal{D}_β

Ein Modell für die Typenlogik \mathcal{M} besteht dann ebenfalls aus einer nicht-leeren Menge \mathcal{D} und einer Abbildung F , die jeder Konstanten vom Typ α ein Element von \mathcal{D}_α zuordnet.

„Hans schläft und Hansi schläft nicht.“

$$\{ \text{Hans}, \text{Hansi} \} = \mathcal{D}_e$$

$$F(\text{Hans}) = \text{„Hans“} \in \mathcal{D}_e$$

$$F(\text{Hansi}) = \text{„Hansi“} \in \mathcal{D}_e$$

$$F(\mathbf{Schlafen(x)}) = \mathcal{D}_{\langle e, t \rangle}$$

$$\mathcal{D}_{\langle e, t \rangle} = \{ \langle \text{Hans}, 1 \rangle, \langle \text{Hansi}, 0 \rangle \}$$

Eine Erweiterung des Mechanismus der Typenlogik bietet der λ -**Kalkül**. Dies erscheint notwendig, da z. B. aus der Aussage „*alle arbeiten*“ Probleme entstehen. In der Typenlogik kann keine Aussage über die Mengenbezeichnung *alle* gemacht werden und somit nicht über Mengen von Elementen in \mathcal{D} . Dies wird durch den λ -Operator erreicht, der eine Variable genau wie eine Quantor bindet, z. B. λx (**Arbeiten**(x)). Alle syntaktischen Kategorien (z. B. NP und VP) müssen einem semantischen Typ zugeordnet werden. Daher existiert ein *semantisches Lexikon*, das Wörter logischen Ausdrücken zuordnet. Zusätzlich existieren die *semantischen Regeln*, die zu jeder syntaktischen Regel eine semantische Bedeutung angeben. Dadurch kann eine semantische Interpretation eines Satzes erfolgen.

Ambiguitäten können mit dieser Beschreibung zwar dargestellt, aber nicht aufgelöst werden. Das Hauptproblem bleibt allerdings der variable Skopus. So ist z. B. aus dem Satz „*Jeder Student hat ein Problem.*“ nicht eindeutig zu erkennen, ob es sich um ein spezielles oder ein allgemeines Problem handelt. Dabei kommt der Umstand zum tragen, dass die oben beschriebenen Mechanismen Aufgaben der Semantik der Syntax zuordnen. Ein Verfahren, das die Skopusproblematik behandelt, ist z. B. der *Cooper-Skopus* [GRS00, S. 753ff]. Dieser betrachtet Skopuseigenschaften, die von den syntaktischen Strukturen abweichen. Allerdings entsteht dabei das Problem, dass die Anzahl der möglichen Kombinationen von Quantorverbindungen zu groß wird.

Die Analysen der vorhergehenden Methoden sind strikt satzgebunden. Allerdings gibt es viele Probleme, so benötigt z. B. ein Personalpronomen in einem Satz einen Bezug auf den Kontext des vorhergehenden Satzes. Mit Satzsemantiken lässt sich dieses Problem nicht lösen. Dazu müssen Methoden der **Diskurssemantik** verwendet werden.

Beispiel 2.4.6 (Anaphorische Beziehungen)

Eine anaphorische Beziehung ist ein referentieller Verweis auf einen vorher geäußerten sprachlichen Ausdruck.



Abbildung 2.27: Beispiel einer anaphorischen Beziehung

Die **Diskursrepräsentationstheorie** (DRT) [Kam81, Hei83] betrachtet daher nicht nur einzelne Sätze, sie besitzt Mechanismen, die anaphorische Beziehungen nachvollziehen können. So existieren in der Diskursrepräsentationstheorie *Diskursrepräsentationsstrukturen*, die aus zwei Komponenten bestehen, nämlich aus Objekten, die sich auf anaphorische Pronomina beziehen (*Menge von Diskursreferenten*), und aus einer Menge von Bedingungen, die für die Diskursreferenten die Wahrheitsbedingungen festlegen.

Beispiel 2.4.7

„**Der Hund hat Fleisch. Er frisst es.**“

Die Diskursreferenten sind x und y .

1. Bedingung: $\{x, y \mid (x = \text{Hund}) \text{ und } (y = \text{Knochen}); (x \text{ hat } y)\}$
2. Bedingung: $\{x, y \mid (x = \text{er}); (y = \text{es}); (x \text{ frisst } y)\}$

In diesem Fall ist die Auswahl der Diskursreferenten einfach, da der Hund männlich ist und der Knochen sächlich. Normalerweise ist die Auswahl der Diskursreferenten jedoch nicht so leicht. Die syntaktische Analyse wird durch Konstruktionsregeln in die Diskursrepräsentationsstrukturen, Diskursreferenten und in die Bedingungen zerlegt, so dass eine semantische Interpretation stattfinden kann.

Erweiterungen dieses Ansatzes sind z. B. die *dynamische Prädikatenlogik* [GS91] und die λ -DRT [KKP96].

Die semantische Analyse der einzelnen Wörter erfolgt anders als die der Satzsemantik und der Diskurssemantik. Es gibt zahlreiche Methoden, um die Wortsemantik zu analysieren. Wortbedeutungen werden z. B. als strukturierte Objekte dargestellt, die durch Regeln abgeleitet werden [CB95]. Lexikalische Einheiten erhalten unterspezifizierte semantische Repräsentationen, die allen Lesarten gemein sind und zusätzlich Regeln, die auf pragmatische Anteile Bezug nehmen.

Pragmatik. Die informationstechnische Realisierung der Semantik orientiert sich an der wahrheitskonditionalen Semantik. Durch die Angabe von Wahrheitsbedingungen wird der Informationsgehalt extrahiert. Alle anderen bedeutungsrelevanten Betrachtungen werden nicht untersucht, wie z. B. sprachliches Handeln. Dies wird durch Mittel der *Pragmatik* untersucht. Untersuchungsgegenstände sind Deixis, Präsuppositionen usw. [CEE⁺01, S. 306].

Präsuppositionen sind Wissensvoraussetzungen, die ein menschlicher Leser während des Lesens annimmt. *„Es gibt in jeder Sprache Wörter, die weniger einen semantischen Gehalt als vielmehr eine Zeigefunktion haben, die so genannten Deiktika. Hierzu gehören die Wörter wie hier und dort, jetzt und dann. Es sind Wörter, die nicht z. B. einen absoluten Ort oder Zeitpunkt angeben, sondern einen solchen bezogen auf Sprecher/Schreiber oder Hörer/Leser.“* [Dud98, 848]

Diese Betrachtungsweisen treten im allgemeinen in der Mathematik nicht auf und werden daher im nächsten Schritt nicht weiter behandelt.

2.4.4 Fachsprache

Fachsprache ist ein weitgefasster Begriff und beschreibt einen speziellen Sprachstil. *„Zunächst ist fraglich, ob bei dem Gegenstand, den man mit Fachsprache meint, die Fachlichkeit wirklich das primäre ist, oder nicht vielmehr textsortenspezifische, semantische, pragmatische oder sonstige Argumente. Zum zweiten, nehmen wir einmal an, man wollte an der Fachlichkeit grundsätzlich festhalten, ist die Frage unklar, was eigentlich Fächer konstituiert, ob z. B. Naturwissenschaft ein Fach ist, oder Chemie, oder nur die organische Chemie oder gar nur die Chemie der Aminosäuren.“* [vH83, S. 63]

„Fachsprachen sind erstaunlich vielseitig. Sie unterscheiden sich nicht nur nach Fachbereichen oder im Hinblick darauf, ob sie gesprochen oder geschrieben werden. Man muss auch die innere Differenzierung berücksichtigen, den jeweiligen Grad der Fachsprachlichkeit, die verschiedenen Textsorten innerhalb einer Fachsprache, die kulturellen Unterschiede und Konventionen fachlicher Kommunikation.“ [GHL02, S. 613] So wird in Lehrbüchern eine andere Form der Fachsprache geschrieben als in wissenschaftlichen Aufsätzen.

Trotzdem enthält z. B. die schriftliche Textform häufig charakteristische Merkmale. So werden Sätze gerne *„durch Zahlen oder Formeln, Abbildungen, Schemata und Tabellen unterbrochen“* [GHL02, S. 618], oder es werden Spiegelstriche bzw. Klammern verwendet. Typographische Mittel sind z. B. die Hervorhebung von Wörtern durch Fett- oder Kursivdruck. Im sprachlichen Bereich werden viele Verweise untereinander gemacht, so dass der Text kohärent wirkt. Dazu werden häufig Artikelwörter (z. B. „dieses“), Pronominaladverbien (z. B.

„darauf“), feststehende Phrasen (z. B. „im Folgenden“), logische Verknüpfungen durch nebenordnende Konjunktionen (z. B. „ferner“), Quantifizierungen durch Gradpartikel (z. B. „wenige“), häufige Wiederholungen von Fachbegriffen usw. verwendet [GHL02].

Jede Fachsprache besitzt einen für sie typischen Wortschatz, der ein Schlüssel zur Kommunikation in der Fachsprache ist. Wie jedoch oben schon erwähnt wurde, ist es schwierig Fachsprachen einzugrenzen. So gibt es in der Mathematik sehr unterschiedliche Fachsprachen mit unterschiedlichen Wortschätzen. Allerdings werden die Fachwörter in eindeutiger Weise definiert, so dass die Kommunikation wesentlich einfacher wird als in anderen Fachsprachen anderer Gebiete. Vorsicht ist jedoch dabei geboten, die Fachwörter auf die präzisen Ausführungen in der Fachsprache zu übertragen. So ist die Verwendung von Gradpartikeln ein eindeutiges Zeichen für die Verwendung von nicht eindeutigen Beschreibungen. Des Weiteren darf aus dem Stamm von Fachwörtern nicht auf ihre tatsächliche Bedeutung geschlossen werden. So gibt es z. B. Begriffe, die zunächst den Anschein erwecken, zu einer bestimmten Familie von Begriffen zu gehören. Im weiteren Verlauf kann sich aber herausstellen, dass dem nicht so ist.

In der Syntax gibt es bei den Fachsprachen einige charakteristische Besonderheiten. Die Satzstrukturen sind *„einfacher, kürzer, klar überschaubar gegliedert“*, aber es werden *„die Satzglieder immer länger, weil die Nomina mit Attributen oder Komposita angereichert“* [GHL02, S. 622] werden. Generell können folgende syntaktische Strukturen gefunden werden (Auswahl aus [GHL02, s. 622]):

- Verwendung von Nomina und nominalen Ausdrücken;
- Verwendung von Satzgliedern anstelle von Nebensätzen;
- Verwendung von erweiterten Attributen anstelle von Attributsätzen;
- Verwendung von substantivierten Verben;
- Verwendung von Präpositionalgefügen statt Vollverben;
- Verwendung von bestimmten Verben: *ist, scheint, zeigt sich*;
- Verwendung von Infinitiv- und Passivkonstruktionen: *„lässt sich zeigen“*;
- Verwendung von Depersonalisierungen: *„man definiert“*;

- Verwendung von Konditionalsätzen: „*wenn ... , dann*“;
- Verwendung von Finalsätzen;

2.5 Information Retrieval

Mache die Dinge so einfach wie möglich — aber nicht einfacher.
(Albert Einstein)

Nach Rijsbergen [vR79, S. 3] besteht folgendes Problem: „*Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by his question. He can obtain the set by reading all the documents in the store, retaining the relevant documents and discarding all the others. In a sense, this constitutes 'perfect' retrieval. This solution is obviously impracticable. A user either does not have the time or does not wish to spend the time reading the entire document collection, apart from the fact that it may be physically impossible for him to do so.*“

Information Retrieval beschäftigt sich mit Verfahren zur Darstellung, Speicherung und Organisation von sowie dem Zugriff auf Informationen. Es werden „*Informationssysteme in Bezug auf ihre Rolle des Wissenstransfers vom menschlichen Wissensproduzenten zum Informations-Nachfragenden betrachtet.*“ [fl04] Information Retrieval Systeme dienen dem einfachen Zugang zu Informationen für Nutzer eines Systems [BYRN99, S. 1]. Daten sind hauptsächlich natürlich-sprachliche Texte, die informationstechnisch unstrukturiert und ambig sind (Kapitel 2.4).

Textdokumente werden z.B. in Datenbanken gespeichert und organisiert. In einem Information Retrieval System werden diese Informationen in eine interne Repräsentation überführt, die das System effizient verarbeiten kann. Dabei wird ein Index erzeugt (Indextermgenerierung), der das Dokument durch Indexterme darstellt. Dieser Prozess verhält sich ähnlich wie die Unix-Befehle `updatedb` und `locate`. `updatedb` verwendet Methoden, um Informationen über das Filesystem in eine einfach zu durchsuchende Repräsentation (z. B. Liste) abzubilden. Mittels des Befehls `locate` können diese Informationen schnell abgerufen werden.

Anfragen des Nutzers (*queries*) an das System werden ebenfalls in eine interne Repräsentation überführt, um sie dem System zugänglich zu machen. Der Prozess der Suche verbindet nun den Index mit der Abfrage und generiert mittels geeigneter Methoden Dokumente, die der Abfrage entsprechen. Ein nachfolgen-

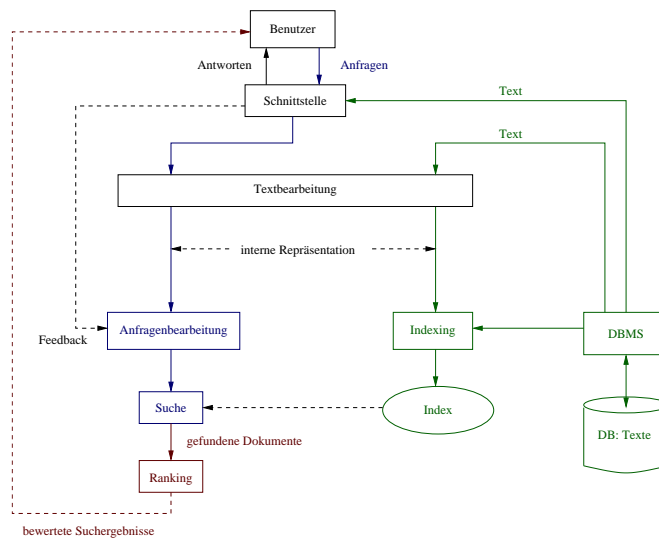


Abbildung 2.28: Architektur eines Information Retrieval Systems nach [BYRN99]

des Rankingsystem sortiert dann die gefundenen Dokumente nach ihrer Relevanz für die Benutzeranfrage und liefert sie als bewertete Suchergebnisse an den Nutzer zurück.

Es existieren zahlreiche Methoden, um möglichst effizient Antworten auf Anfragen der Nutzer eines Information Retrieval Systems zu generieren. Klassische Methoden wie das *Boolesche Information Retrieval* verwenden die Boolesche Algebra. Es ist ein sehr einfacher Mechanismus, der Dokumente nur in relevant oder nicht-relevant unterteilt. Eine weitere klassische Methode ist das *Vektorraum Information Retrieval*. Dieses weist den Indextermen in Anfragen und Dokumenten ein nicht-binäres Gewicht zu. Diese Gewichte werden genutzt, um den Ähnlichkeitsgrad (*Degree of Similarity*) zu berechnen, nach denen die Informationen sortiert werden. Das *probabilistische Information Retrieval* sucht mittels wahrscheinlichkeitstheoretischer Methoden nach idealen Antworten.

Neben den klassischen Methoden existieren einige abgeleitete alternative Modelle wie z. B. das *Fuzzy Information Retrieval* [OMK91] als Alternative zum Booleschen Modell, das *verallgemeinerte Vektormodell* [WZW85], das *Latent Semantic Indexing Modell* [DDF⁺90] und das *Neuronale Netzwerkmodell* als Al-

alternativen zum Vektorraummodell sowie *Inferenznetzwerkmodelle* [Pea88] und *Bayesian Belief Modelle* [BYRN99] als Alternativen zum probabilistischen Modell.

Die *strukturierten Text Retrieval Modelle* ermöglichen dem Nutzer, komplexere Anfragen an das Information Retrieval System zu stellen, im Gegensatz zu den klassischen Methoden und ihren Erweiterungen, die nur Zeichenketten in ihren Anfragen zulassen. Neben den Zeichenketten können zusätzlich bei dieser Methode strukturelle Umgebungsinformationen wie Positionen, Labeleigenschaften usw. übergeben werden, die dann von weiteren Analyseschritten genutzt werden können. Die *Non-Overlapping List* Methode [Bur92] zerlegt einen Text auf verschiedene Weisen in nicht-überlappende Regionen und legt diese in einer oder mehreren Listen ab. Die *Proximal Nodes* Methode [NBY97] legt Textstrukturen in einer hierarchischen Struktur und einem einfachen Wortindex ab.

Durch die Methode des *Browsing* besteht die Möglichkeit, durch Kommunikation zwischen Nutzer und System Suchanfragen zu verändern und anzupassen. Hierbei wird davon ausgegangen, dass die Ziele des Nutzers nicht von Anfang an bestimmbar sind. Browsing existiert in den drei Varianten *flat*, *structure guided* und *Hypertext*. Beim *flat browsing* wird dem Benutzer das Suchergebnis unstrukturiert angezeigt, z. B. als Dokumentenliste. Anhand dieses Überblicks kann der Benutzer dann seine ursprüngliche Suchanfrage modifizieren, indem z. B. die Suchkriterien eingeschränkt oder erweitert werden. Dieser Prozess wird auch als *relevance feedback* bezeichnet. Um das Browsing zu erleichtern, wird beim *structure guided browsing* anstelle einer flachen Übersicht über die Dokumente eine hierarchische Struktur gewählt. Das *Hypertext Modell* ergänzt die existierenden Dokumentenstrukturen durch eine zusätzliche, dokumentenübergreifende Navigationsebene.

Eine formale Charakterisierung für ein Information Retrieval Modell wird gegeben durch ([BYRN99], S. 23):

Definition 2.5.1 (Information Retrieval Modell)

Ein **Information Retrieval Modell** ist ein Quadrupel $(D, Q, \mathcal{F}, R(q_i, d_j))$ mit:

1. D ist eine Menge von Repräsentationen der zu betrachtenden Dokumente.
2. Q ist eine Menge von Repräsentationen von Benutzeranfragen. Solche Repräsentationen werden auch als *Queries* bezeichnet.

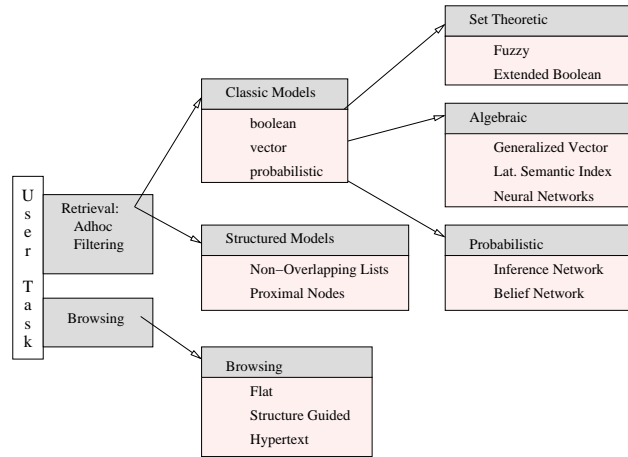


Abbildung 2.29: Taxonomie von Information Retrieval Modellen [BYRN99]

3. \mathcal{F} ist ein System zur Modellierung von Dokumentenrepräsentationen, Queries und ihren Abhängigkeiten.
4. $R(q_i, d_j)$ ist eine **Ranking Funktion** (Bewertungsfunktion), die von $Q \times D$ nach \mathbb{R} abbildet. Diese Bewertung definiert eine Ordnung der Dokumente $d_j \in D$ in Bezug auf einen Query $q_i \in Q$.

Jedes Dokument wird durch eine Menge von Schlüsselwörtern (*index terms*) beschrieben, die durch ihre Semantik das Dokument charakterisieren (Definition 2.5.2). Dabei ist es möglich, jeden Indexterm in Bezug auf verschiedene Dokumente unterschiedlich zu gewichten (Definition 2.5.3).

Definition 2.5.2 (Indexterm)

„An index term (keyword) is a pre-selected term which can be used to refer to the content of a document. Usually, index terms are nouns or noun groups. In the Web, however, some search engines use all the words in a document as index terms.“ [BYRN99, S. 444]

Definition 2.5.3 (Gewichtung)

Sei $K = \{k_1, \dots, k_t\}$ die Menge aller Indexterme und D die Menge aller Dokumente. Dann heißt der Vektor $\vec{w}_j = (w_{1,j}, \dots, w_{t,j}) \in \mathbb{R}^t$ **Indextermvektor**

des Dokumentes $d_j \in D$ mit den Gewichtungen

$$w_{i,j} = \begin{cases} > 0 & \text{falls } k_i \text{ in } d_j \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$$

Weiterhin definieren wir eine Funktion $g_i : \mathbb{R}^t \rightarrow \mathbb{R}$ mit $g_i(\vec{w}_j) = w_{i,j}$ (nach [BYRN99, S. 25]).

Üblicherweise wird vereinfachend davon ausgegangen, dass die Gewichtungen $w_{i,j}$ paarweise unkorreliert sind.

2.5.1 Methoden des Information Retrieval

Boolesches Information Retrieval. Beim booleschen Information Retrieval besteht eine Anfrage aus der Verknüpfung von atomaren Anfragen. Eine atomare Anfrage prüft dabei, ob ein Indexterm in einem Dokument vorkommt. Mit Hilfe der Booleschen Operatoren — AND, OR und NOT — können dabei aus einfachen Anfragen komplexe Ausdrücke zusammengesetzt werden (Kapitel 3.2.2).

Definition 2.5.4 (Boolesches Information Retrieval Modell)

Für das boolesche Modell sind die Gewichte der Indexterme binär, $w_{i,j} \in \{0, 1\}$. Eine Anfrage q ist ein boolescher Ausdruck. Sei \vec{q}_{dnf} die disjunkte Normalform [Spi04, S. 51] von q . Des Weiteren, sei \vec{q}_{cc} eine beliebige konjunktive Komponente von \vec{q}_{dnf} . Die Ähnlichkeit eines Dokumentes d_j zu der Abfrage q ist definiert als:

$$\text{sim}(d_j, q) = \begin{cases} 1 & \text{falls } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{w}_j) = g_i(\vec{q}_{cc})) \\ 0 & \text{sonst} \end{cases}$$

Für $\text{sim}(d_j, q) = 1$ sagt das Boolesche Modell aus, dass das Dokument d_j relevant zu der Abfrage q ist. Sonst ist das Dokument nicht relevant. (nach [BYRN99, S. 26])

Das bedeutet jedoch, dass ein Dokument nur dann als relevant erkannt wird, wenn die Anfrage vollständig erfüllt ist. Eine teilweise Übereinstimmung (*partial matching*) ist daher nicht möglich. Ein solches System lässt sich von ungeübten Nutzern nur schwer bedienen. Daher werden meistens zwei verschiedene Interfaces für einfache bzw. erweiterte Suche angeboten.

Fuzzy Information Retrieval. Das Fuzzy Information Retrieval basiert auf der Theorie der Fuzzy-Mengen, verwendet aber trotzdem noch Methoden des Booleschen Information Retrieval. Die Grundidee dieses Konzeptes ist es, Mengenzuordnungen nicht mehr eindeutig Objekten zuzuordnen, sondern einer Merkmalsklasse. Somit werden einem Dokument keine eindeutigen Indexterme, sondern korrelierte Terme zugeordnet. Damit können auch Begriffe gefunden werden, die eng mit dem zu suchenden Begriff im Dokument in Beziehung stehen.

Bei der Indexierung wird eine Korrelationsmatrix zwischen den Indextermen konstruiert. Das Retrieval besteht darin, die Dokumente mit dem höchsten Grad an Zugehörigkeit zu den in der Anfrage vorkommenden Termen oder ihren korrelierten Termen zu ermitteln [BYRN99, S. 35ff].

Erweitertes Boolesches Information Retrieval. Das Erweiterte Boolesche Information Retrieval ergänzt die klassische boolesche Methode durch die Einbeziehung von teilweise erfüllten booleschen Ausdrücken. Dadurch wird das „Alles-oder-Nichts“-Problem der klassischen Methode vermieden. Die Bewertung erfolgt mit Hilfe von l_p -Normen, wobei p durch den Benutzer bei der Anfrage einstellbar ist. [BYRN99, S. 38ff]

Vektorraum Information Retrieval Das Vektorraum Information Retrieval ist ein Modell, das *partial matching* zulässt. Alle Indexterme erhalten reelle Gewichtungsfaktoren für alle Dokumente und die Anfrage. Diese Gewichte werden genutzt, um die Ähnlichkeit zwischen einem Dokument und der Anfrage zu berechnen.

Definition 2.5.5 (Vektorraum Information Retrieval Modell)

Für ein Vektorraum Information Retrieval Modell wird jedem Paar (k_i, d_j) mit $k_i \in K$, $d_j \in D$ ein Gewichtungsfaktor $w_{i,j} \geq 0$ zugeordnet. Jedem Indexterm k_i wird in der Anfrage q ebenfalls ein Gewicht $w_{i,q} \geq 0$ zugeordnet. Dann ist der Anfragevektor \vec{q} definiert als $\vec{q} = (w_{1,q}, \dots, w_{t,q})$ und der Dokumentenvektor von d_j definiert durch $\vec{w}_j = (w_{1,t}, \dots, w_{j,t})$. (nach [BYRN99, S. 27])

Aus \vec{q} und \vec{w}_j kann nun der Grad der Ähnlichkeit $\text{sim}(d_j, q)$ zwischen der Anfrage q und dem Dokument d_j berechnet werden. Eine mögliche Maß dafür ist

das Kosinusmaß:

$$\text{sim}(d_j, q) = \frac{\langle \vec{w}_j, \vec{q} \rangle}{|\vec{w}_j| |\vec{q}|}$$

Es existieren zahlreiche Verfahren für die Bestimmung der Indextermgewichte $w_{i,j}$. Die bekanntesten verwenden dabei so genannte *tf-idf-schemes*, die sich an der Häufigkeit der Indexterme orientieren:

$$\begin{aligned} w_{i,j} &= \frac{\text{freq}_{i,j}}{\max_l \text{freq}_{l,j}} \log \frac{N}{n_i} \\ w_{i,q} &= \left(0.5 + \frac{0.5 \text{freq}_{i,q}}{\max_l \text{freq}_{l,q}} \right) \log \frac{N}{n_i} \end{aligned}$$

Dabei ist N die Anzahl der Dokumente, n_i die Anzahl der Dokumente, in denen der Indexterm k_i vorkommt, und $\text{freq}_{i,j}$ bzw. $\text{freq}_{i,q}$ ist die Häufigkeit des Indexterms k_i im Dokument d_j bzw. in Query q .

Das Vektorraum Information Retrieval Modell ist ein sehr leistungsfähiges Verfahren, das sich in der Praxis als nur schwer zu schlagen erwiesen hat. Dies und seine Einfachheit sowie die hohe Geschwindigkeit haben es zu einem der am meisten verwendeten Verfahren gemacht [BYRN99, S. 27].

Verallgemeinertes Vektorraum Information Retrieval Modell. Das Verallgemeinerte Vektorraum Information Retrieval Modell ist eine Erweiterung des klassischen Vektorraum Modells, bei dem zusätzlich die Korrelation zwischen den Indextermen berücksichtigt wird. Dabei ist nicht klar, ob die Verwendung dieser Korrelation zu einer Verbesserung des Ergebnisses führt [BYRN99, S. 41].

Probabilistisches Information Retrieval. Das probabilistische Information Retrieval verwendet wahrscheinlichkeitstheoretische Methoden zur Bestimmung der Wahrscheinlichkeit, dass ein Dokument als relevant eingestuft wird.

Definition 2.5.6 (Probabilistische Information Retrieval Modell)

Für das probabilistische Modell sind die Gewichtungen der Indexterme binär, d. h. $w_{i,j} \in \{0, 1\}$, $w_{i,q} \in \{0, 1\}$. Eine Abfrage q ist eine Untermenge von Indextermen. Sei R_q die Menge der für q relevanten Dokumente. Sei \bar{R}_q das Komplement von R_q (Menge der nichtrelevanten Dokumente). Sei $P(R_q \mid \vec{w}_j)$ die

Wahrscheinlichkeit, dass das Dokument d_j relevant für die Abfrage q ist und $P(\bar{R}_q \mid \vec{w}_j)$ die Wahrscheinlichkeit, dass das Dokument d_j nicht relevant für q ist. Die Ähnlichkeit $\text{sim}(d_j, q)$ des Dokumentes d_j zu der Abfrage q ist definiert durch:

$$\text{sim}(d_j, q) = \frac{P(R_q \mid \vec{w}_j)}{P(\bar{R}_q \mid \vec{w}_j)}.$$

(nach [BYRN99, S. 32])

Kapitel 3

Mathematische Strukturen

E: Zu welchen Menschen gehören die Lernenden, zu den Klugen oder zu den Dummen?

K: Zu den Klugen.

E: Aber solange ihr noch lerntet, habt ihr noch nicht gewusst, was ihr lerntet; wart ihr also klug, als ihr das noch nicht wusstet?

K: Nein.

E: Also wart ihr dumm, als ihr lerntet.

K: Ich muss es zugeben.

E: Wer von euch lernte, was der Lehrer euch sagte, die Klugen oder die Dummen?

K: Die Klugen.

E: Also lernen doch die Klugen und nicht die Dummen [...].

(Platon; Euthydemos)

3.1 Einleitung

Einen Überblick über die Mathematik in Teilbereichen oder sogar als Ganzes zu erlangen, ist ein schwieriges Problem. Die Mathematik ist eine sehr komplexe Wissenschaft, in der verschiedene Spezialgebiete nebeneinander existieren. In einigen Teilbereichen wird hingegen versucht, über *Abstraktionen* von bekannten Spezialgebieten, neue Theorien zu entwickeln. Ein Beispiel für eine solche Abstraktion ist der Zusammenschluss der Analysis und der Geometrie

zur Differentialgeometrie. Bei solch einem komplexen Gebiet sind selbst hervorragende Mathematiker heute stets nur Spezialisten auf einem und einigen wenigen Teilbereichen und besitzen nur einen oberflächlichen Überblick über die Mathematik als Ganzes. Auch innerhalb einer Fachdisziplin gibt es zahlreiche Zusammenhänge und Verknüpfungen, die nur schwer zu überblicken sind.

Es wäre vorteilhaft, wenn eine systematische Darstellungsform der Mathematik gefunden werden könnte, die mathematisches Wissen zumindest teilweise kompakt darstellt. Es handelt sich dabei um eine *Taxonomie der Mathematik*, d. h. die Einordnung von mathematischen Begriffen in ein semantisches Netzwerk (Kapitel 2.2). Dabei ist es unwahrscheinlich, dass eine vollständige Darstellung der gesamten Mathematik realisiert werden kann. Daher müssen Teilausschnitte der Mathematik so konstruiert werden, dass sie dann in geeigneter Weise zusammenfügbar sind. Probleme, die bei Wissensdarstellungen auftreten, wurden im Kapitel des Wissensmanagements (Kapitel 2.2) diskutiert.

Die Mathematik als Untersuchungsobjekt bietet zahlreiche Vorteile. Durch ihre knappe und exakte Ausdrucksweise bietet sie sich an, Wissen maschinell zu extrahieren. Diese Ausdrucksweise bezeichnet man als *axiomatische Darstellung* oder *Beschreibung*. In der axiomatischen Darstellung werden aus Grundaussagen, den *Axiomen*, weitere Aussagen entwickelt. Die Axiome bleiben dabei unbewiesen und werden als wahr angenommen. Durch diesen Ansatz wird ein formales Grundgerüst bereitgestellt, das eine mathematische Theorie in strukturierte Spracheinheiten zerlegt und das mathematische Wissen partitioniert. Zum Beispiel sind die *Definitionen* solche Spracheinheiten. Sie determinieren den terminologischen Rahmen der mathematischen Betrachtungen und vereinfachen somit Strukturen. Weitere Spracheinheiten bilden die Sätze. Sie enthalten wahre Aussagen über mathematische Sachverhalte, die durch spezielle logische Regeln erzeugt werden. Beweise begründen neue Sätze auf Grundlage der schon vorhandenen Sätze und Axiome mithilfe von logischen Schlussregeln.

Der entscheidende Vorteil entsteht durch die grammatikalischen Konstruktionen der mathematischen Sprache. Die Sätze werden meistens nach bestimmten Mustern gebildet, so dass eine endliche Anzahl von Phrasen existiert, die immer wieder verwendet werden. Die mathematische Sprache ist außerdem eine reduzierte Sprache, die sehr viel weniger Ausdrucksmöglichkeiten zulässt als die Alltagssprache.

Auch ein menschlicher Leser muss die mathematische Sprache wie eine Fremdsprache erlernen. Sie hat mit den Alltagssprachlichen Texten kaum Gemeinsamkeiten. Allerdings findet in den letzten Jahrzehnten eine Mathematisierung der Alltagssprache statt, die teilweise bedingt wird durch Verwendung von technischen Begriffen, die aus der Informationstechnik stammen.

Die mathematische Sprache bildet den Schlüssel zum Verständnis der Mathematik. Wer die Semantik dieser Sprache nicht versteht, kann weder die Konzepte verstehen noch mit Mathematikern kommunizieren. Dabei geht es nicht nur um das Verständnis der einzelnen Formeln und Symbole, sondern auch um einfach konstruierte natürlichsprachliche Texte.

In diesem Kapitel werden insbesondere zwei Aspekte unterschieden: Es sollen auf der einen Seite die Sprachstrukturen (Kapitel 3.3) und auf der anderen Seite die Wissensstrukturen (Kapitel 3.4) in der Mathematik betrachtet werden. Damit verbunden sind die Fragen, wie sich die mathematische Sprache analysieren lässt und wie mathematisches Wissen sinnvoll dargestellt werden kann. Um diese Fragen beantworten zu können, müssen die grundlegendsten Konzepte der Mathematik — mathematische Logik (Kapitel 3.2.2) und axiomatische Mengenlehre (Kapitel 3.2.3) — zuallererst betrachtet werden.

3.2 Grundlagen

Man muss viel studiert haben, um wenig zu wissen.
(Montesquieu, franz. Schriftsteller, 1689–1755)

3.2.1 Einleitung

Die Mathematik ist in ihren Formulierungen strenger und genauer als andere Fachsprachen (Kapitel 2.4.4). Grundlage der Strenge und Genauigkeit sind die logische Betrachtungsweise und der axiomatische Aufbau der Mathematik.

Die ersten axiomatischen Beschreibungen der Mathematik entstanden schon in der Antike. So beschrieb Euklid (350–275 v. Chr.) in dem Buch „*Die Elemente*“ den axiomatischen Aufbau der Geometrie. Seit Ende des 19. Jahrhunderts gab es zahlreiche Versuche, eine axiomatische Betrachtungsweise auf die ganze Mathematik zu übertragen (*moderne Mathematik*). Durch logische Paradoxien wurde allerdings das Bild der Mathematik als über alle Zweifel erhabene Wissenschaft massiv gestört (*Grundlagenkrise der Mathematik*).

Schon durch Immanuel Kant (1724–1804) wurden die ersten Grundlagen für die Sichtweise der modernen Mathematik festgelegt. So unterschied er zwischen analytischen und synthetischen Aussagen:

- analytisch wahre Aussage: Das Prädikat enthält nur, was im Subjekt bereits enthalten ist. Analytische Aussagen vermitteln demnach kein neues Wissen. Die Aussage *Alle Junggesellen sind unverheiratet* ist wahr, weil der Begriff *Junggeselle* unverheirateter Mann bedeutet.
- synthetische Aussage: Hier enthält das Prädikat immer etwas neues, im Subjekt noch nicht enthaltenes. Damit vermitteln synthetische Aussagen neues Wissen.

Synthetische Aussagen lassen sich nochmals unterteilen in *Aussagen a priori* und *Aussagen a posteriori*. Die synthetischen Aussagen a priori enthalten dabei allgemeingültige Wahrheiten. Dazu gehören auch mathematische Sätze, die nicht empirisch, sondern durch korrektes Schlussfolgern abgeleitet werden. Allerdings berücksichtigte Kant trotz seiner strikten Ausführungen nicht den

informellen Charakter der menschlichen Sprache. So interpretieren Menschen Sprache normalerweise mithilfe ihres inneren Weltbildes. Da dieses sich von Mensch zu Mensch unterscheiden kann, ist eine solche Interpretation für exakte wissenschaftliche Aussagen ungeeignet. So beschreibt z. B. der unten stehende Satz des deutschen Logikers Kurt Grelling den Fall von semantischen Widersprüchen.

„Einige Adjektive wie etwa kurz und deutsch treffen auf sich selbst zu, andere etwa lang und englisch, nicht. Wir nennen jene der ersten Gruppe autologisch und jene der zweiten Gruppe heterologisch. Ist das Adjektiv heterologisch selbst heterologisch? Falls ja, trifft es nach dieser Definition nicht auf sich selbst zu und muss heterologisch sein.“ ([Bar94, S. 170])

Diese Erkenntnis führte zur Entwicklung einer exakten mathematischen Sprache auf Basis der Logik und somit zur Trennung von Syntax und Semantik.

Einer der bekanntesten Wegbereiter ist David Hilbert (1884–1943), der den Formalismus der Mathematik als ein *strategisches Spiel* mit Zeichenreihen sah, dessen Spielmaterial die Axiome und dessen Spielregeln die Schlussregeln waren. Die gesamte Mathematik sollte auf Axiomen und mathematischen Grundaussagen aufgebaut werden, ein Kalkül formalisiert die Logik und erlaubt es jede korrekte mathematische Aussage herzuleiten (Syntax) und Aussagen auf ihren Wahrheitsgehalt zu überprüfen (Semantik). Das Kalkül sollte konsistent sein und nur wahre Aussagen ableitbar machen. Und aus ihm sollten alle wahren Aussagen herleitbar sein (Vollständigkeit). Dieser Plan wird als *Hilbert Programm* bezeichnet.

Die Grundobjekte der mathematischen Sprache bilden die Mengen, die durch Georg Cantor (1845–1916) eingeführt wurden. Ein formales System ist u. a. durch Russell und Whitehead [WR84] gegeben (*Typentheorie*). Des Weiteren gibt es das *Zermelo-Fraenkelsche Axiomensystem*, aus dem die axiomatischen Ableitungen aller Lehrsätze der klassischen Analysis hergeleitet werden können.

Die Grundlage aller Beschreibungen der Mathematik liefert die mathematische Logik. Dies ist die *Prädikatenlogik 1. Stufe mit Identität* (Kapitel 3.2.2). Ihre Basis ist die *Aussagenlogik*. Die Aussagenlogik beschreibt die Beziehungen,

die zwischen den Aussagen stehen. Die Prädikatenlogik beschreibt zusätzlich die innere Struktur der Aussagen mittels Quantoren. Wissenschaftlich gesehen ist die mathematische Logik eine Sprache (Kapitel 2.4), die durch eine Syntax und Semantik charakterisiert wird. Darüberhinaus spielt für die Technik der Beweisführung und für die Konsistenz einer mathematischen Logik die Frage der Entscheidbarkeit eine bedeutende Rolle (Gödel [Gö31]).

3.2.2 Mathematische Logiken

Die mathematische Sprache ist eine formalisierte künstliche Sprache (Kapitel 3.3) auf Basis der natürlichen Sprache und der **mathematischen** bzw. **formalen Logik**. In den Sätzen der mathematischen Sprache sind Aussagen enthalten:

„Aussagen sind sprachliche Objekte zur Mitteilung von Sachverhalten.“ [Rau79, S. 6]

Sätze, die Aussagen enthalten, werden in der natürlichsprachlichen Grammatik als **Aussagensätze** bzw. **Deklarativsätze** bezeichnet. Sie unterscheiden sich von anderen Satzarten wie z. B. Interrogativ-, Desiderativ- oder Exklamativsatz (Kapitel 3.3). So steht beispielsweise im Deklarativsatz das finite Verb im Indikativ oder im Konjunktiv II, aber nie im Imperativ.

Mathematische Aussagen sind gegenüber der natürlichen Sprache strukturierter und setzen sich aus kurzen, prägnanten Satzkonstruktionen zusammen. Dabei bestehen sie aus Grundelementen (atomaren Bausteinen):

„Atomar heißen Aussagen, deren innere Struktur nicht präzisiert wird. In der Umgangssprache entsprechen solche Aussagen am ehesten einfachen Aussagensätzen ohne Nebensätze.“ [Spi04, S. 5]

Einer atomaren Aussage wird eine (Aussage-) Variable, z. B. a, b, c, \dots , zugeordnet. Nach Seiler ([Sei00, I.1.i]) werden mathematische Aussagen folgendermaßen definiert:

Definition 3.2.1

Eine (**mathematische**) **Aussage** ist ein Satz, dem man nach einem abgesprochenen Verfahren genau einen der Wahrheitswerte (bzw. logische Werte einer Aussage) „w“ (wahr) oder „f“ (falsch) zuordnen kann.

Die Definition 3.2.1 beinhaltet ein Axiom, das *Zweiwertigkeitsprinzip*. Es besagt, dass eine Aussage entweder wahr oder falsch sein muss, aber sie kann nicht zugleich wahr und falsch sein. Die *Wahrheitswerte* können formal als eine spezielle zweielementige Menge $\mathbb{B} := \{0, 1\}$ (*boolesche Menge*) dargestellt werden. Dabei steht der Wert 0 für eine falsche Aussage und der Wert 1 für eine wahre Aussage. Die Semantik [Fre94] des Deklarativsatzes reduziert sich auf die Referenz der Wahrheitswerte (Kapitel 2.4).

Mit einzelnen Aussagen kann jedoch keine mathematische Theorie beschrieben bzw. aufgebaut werden. Daher gibt es Operatoren (*Junktoren*), die mathematische Aussagen im Sinne des Extensionalitätsprinzips miteinander verknüpfen. Das *Extensionalitätsprinzip* ist ein weiteres Axiom und bildet mit dem *Zweiwertigkeitsprinzip* die Grundpfeiler einer formalen Sprache der mathematischen Aussagen (**Aussagenlogik**). Das Extensionalitätsprinzip besagt, dass der Wahrheitswert von zusammengesetzten Aussagen sich nur aus den Wahrheitswerten der einzelnen Aussagen zusammensetzt. Diese Verknüpfungen der mathematischen Aussagen werden als *Aussagenverknüpfungen* bzw. *aussagenlogische Verknüpfungen* bezeichnet [Rau79]. Dadurch entsteht die Möglichkeit, komplexe Aussagen aus einfachen Aussagen zu konstruieren.

Satz 3.2.1

Hat man eine oder mehrere Aussagen gegeben, dann kann man mit Junktoren neue Aussagen erzeugen. ([Sei00, I.1.ii])

In der Aussagenlogik werden vor allem fünf Junktoren verwendet. Diese Junktoren können auch als Funktionen beschrieben werden, die als **boolesche Funktionen** $\mathbb{B}^n \rightarrow \mathbb{B}$ ($n > 0$) bezeichnet werden (Abbildung 3.1).

Junktor	Funktion
Negation	$\neg : \mathbb{B} \rightarrow \mathbb{B}$
Konjunktion	$\wedge : \mathbb{B}^2 \rightarrow \mathbb{B}$
Disjunktion	$\vee : \mathbb{B}^2 \rightarrow \mathbb{B}$
Implikation	$\Rightarrow : \mathbb{B}^2 \rightarrow \mathbb{B}$
Äquivalenz	$\Leftrightarrow : \mathbb{B}^2 \rightarrow \mathbb{B}$

Abbildung 3.1: Überblick über die booleschen Funktionen

Diese Verknüpfungen der Aussagen erfolgen nach einem bestimmten Muster, das in den *Wahrheitstafeln* (Abbildung 3.2) festgelegt wird (*Normierung der booleschen Funktionen*).

x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Abbildung 3.2: Wahrheitstafeln

Formale Beschreibung der Aussagenlogik Nun soll die Aussagenlogik als *formale Sprache* dargestellt werden. Zur Beschreibung werden ein Alphabet, eine Syntax und eine Semantik benötigt. Die Syntax beschreibt Sprachen, die aus Zeichen bestehen, wobei verschiedene Typen von Zeichen unterschieden werden. Die Semantik beschreibt die Abbildung (Interpretation) der Syntax auf einer vorgegebenen Struktur. Ein System, das die Syntax und Semantik zusammen mit Schlussfolgerungsregeln beschreibt, wird als *Kalkül* bezeichnet.

Definition 3.2.2 (Alphabet der Aussagenlogik)

Das Vokabular der Aussagenlogik besteht aus einer fest definierten Menge von Symbolen $\Sigma = \{\mathcal{A}, \neg, \Rightarrow, (,)\}$:

1. Ein Alphabet von atomaren Aussagen $\mathcal{A} = \{a, b, c, \dots\}$,
2. Junktoren \neg , \Rightarrow und
3. technische Zeichen (linke und rechte Klammer).

(nach [Spi04, S. 14])

Durch die Verwendung der beiden logischen Operatoren \neg und \Rightarrow können die drei weiteren logischen Operatoren \vee , \wedge und \Leftrightarrow definiert werden:

- $\mathcal{A} \wedge \mathcal{B} := \neg(\mathcal{A} \Rightarrow \neg\mathcal{B})$
- $\mathcal{A} \vee \mathcal{B} := \neg\mathcal{A} \Rightarrow \mathcal{B}$

- $\mathcal{A} \Leftrightarrow \mathcal{B} := \neg(\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow \neg(\mathcal{B} \Rightarrow \mathcal{A})$

Die aus dem Alphabet der Aussagenlogik nach bestimmten Regeln gebildeten Symbolfolgen werden als *aussagenlogische Formeln* bezeichnet. Die Bildungsregeln werden durch die Syntax (Definition 3.2.3) und die Bedeutung durch die Semantik (Definition 3.2.4) beschrieben.

Definition 3.2.3 (Syntax der Aussagenlogik)

Die **aussagenlogischen Formeln** werden folgendermaßen gebildet:

1. Alle atomaren Aussagen aus \mathcal{A} sind aussagenlogische Formeln.
2. Das Negationssymbol \neg gefolgt von einer aussagenlogischen Formel ist eine aussagenlogische Formel.
3. Eine öffnende Klammer $($, gefolgt von einer aussagenlogischen Formel, gefolgt vom Junktoren \Rightarrow , gefolgt von einer aussagenlogischen Formel, gefolgt von einer schließenden Klammer $)$, ist eine aussagenlogische Formel.
4. Bindungsregeln für die Junktoren.

(nach [Spi04, S. 14])

Das durch die Syntax beschriebene formale System kann nun mit einer Semantik verbunden werden:

Definition 3.2.4 (Semantik der Aussagenlogik)

1. Die **Belegung** ist eine Funktion $g : \mathcal{D} \rightarrow \{0, 1\}$, wobei \mathcal{D} die Menge aller aussagenlogischen Formeln ist. Die Belegung wird durch die Wahrheitstafeln (Abbildung 3.2) beschrieben. [CEE⁺01, S. 32]
2. Eine **Interpretation** \mathcal{I} einer aussagenlogischen Formel wird definiert durch:
 - (a) Atomare Aussagen a : $\mathcal{I}(a) = g(a)$;
 - (b) Aussagenlogische Formeln \mathcal{A}, \mathcal{B} :
 - i. Negation: $\mathcal{I}(\neg \mathcal{A}) = 1$, falls $\mathcal{I}(\mathcal{A}) = 0$, sonst $\mathcal{I}(\neg \mathcal{A}) = 0$;
 - ii. Implikation: $\mathcal{I}(\mathcal{A} \Rightarrow \mathcal{B}) = 1$, falls $\mathcal{I}(\mathcal{A}) = 0$ oder $\mathcal{I}(\mathcal{B}) = 1$, sonst $\mathcal{I}(\mathcal{A} \Rightarrow \mathcal{B}) = 0$;

(nach [CEE⁺01, S. 34])

3. Für eine aussagenlogische Formel \mathcal{A} und eine Belegung g schreibt man $g \models \mathcal{A}$, falls $\mathcal{I}(\mathcal{A}) = 1$. Man sagt, **die Formel \mathcal{A} gilt unter der Belegung g** . [CEE⁺01, S. 35]

Die eigentliche Aufgabe der Logik ist es, die Gesetze des logischen Schließens zu untersuchen.

Definition 3.2.5 (Schluss)

Seien nun $\mathcal{P}_1, \dots, \mathcal{P}_n, \mathcal{C}$ aussagenlogische Formeln¹. Wenn $\mathcal{P}_1 \wedge \dots \wedge \mathcal{P}_n \Rightarrow \mathcal{C}$ allgemeingültig ist, nennt man dies einen gültigen Schluss von $\mathcal{P}_1, \dots, \mathcal{P}_n$ nach \mathcal{C} . Man notiert dies durch $\mathcal{P}_1, \dots, \mathcal{P}_n \models \mathcal{C}$. Eine aussagenlogische Formel heißt **allgemeingültig** (Tautologie), wenn sie bei jeder Belegung zu einer Aussage mit Wahrheitswert „wahr“ wird. (nach [Wue95, S. 4, 6])

Die wichtigste Schlussregel ist der *modus ponens* $(\mathcal{P} \Rightarrow \mathcal{C}, \mathcal{P}) \models \mathcal{C}$, d. h., wenn eine Implikation und ihre Prämisse \mathcal{P} angenommen werden, folgt logisch die Konklusion \mathcal{C} .

Um ein vollständiges Beschreibungssystem für die Aussagenlogik zu erhalten, wird ein formales Axiomensystem aufgestellt.

Axiom 3.2.1 (Axiomensystem der Aussagenlogik)

Es seien \mathcal{A}, \mathcal{B} und \mathcal{C} aussagenlogische Formeln. Dann gilt folgendes:

1. aussagenlogische Annahmeneinführung:

$$\mathcal{C} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$$
2. aussagenlogische Verkettungsregel:

$$(\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C}))$$
3. Widerspruchsregel: $(\neg \mathcal{C} \Rightarrow \neg \mathcal{A}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$
4. *modus ponens*

[Spi04, S. 66]²

¹ \mathcal{P} steht für Prämisse und \mathcal{C} für Konklusion.

²System von Lukasiewicz und Tarski

Die Aussagenlogik beschreibt nur einen geringen Teil der Mathematik und reicht deshalb nicht für eine vollständige Beschreibung aus. Eine Erweiterung der Ideen der Aussagenlogik ist durch die *Prädikatenlogik erster Stufe* gegeben. So verwendet die Prädikatenlogik Konzepte der traditionellen Logik wie z. B. die Subjekt-Prädikat Beziehung. Damit können Individuen (Subjekte), deren Eigenschaften (Prädikate) und Geltungsbereiche (Quantoren) abgefragt werden.

„In der Ausdrucksweise der modernen Linguistik entspricht dem Subjekt ein Satzteil, der eine irgendwie geartete Benennung eines Gegenstandes ausdrückt, daher spricht man von einer Nominalphrase. Demgegenüber entspricht dem Prädikat gerade keine Benennung. Linguisten nennen den Satzteil, der grammatisch das Prädikat repräsentiert, die Verbalphrase.“ [Spi04, S. 86]

Des Weiteren existieren Funktoren. Aus linguistischer Sicht können z. B. Funktoren beschrieben werden als Nominalphrasen, die nicht nur durch den Eigennamen eines Individuums angegeben werden, sondern eine funktionale Abhängigkeit beschreiben, z. B. *das Blatt eines Baumes* [Her72].

Nach dieser Betrachtung besteht dann das Alphabet der prädikatenlogischen Sprache aus:

Definition 3.2.6 (Alphabet der Prädikatenlogik)

Das Alphabet der Prädikatenlogik setzt sich zusammen aus dem Alphabet der aussagenlogischen Sprache (Definition 3.2.2) und folgenden Symbolen:

1. *Prädikate*: p, q, \dots ;
2. *Individuenkonstanten*: k, k_1, \dots ;
3. *Individuenvariablen*: x_1, x_2, \dots ;
4. *Funktoren*: f, g, \dots ;
5. *Quantorenzeichen*: Allquantor \forall , Existenzquantor \exists .

Mit dem Alphabet können dann zulässige Symbolfolgen — prädikatenlogische Formeln — gebildet werden, die durch die Syntax der Prädikatenlogik beschrieben werden³.

³Die Bildungsregeln bzw. die Syntax soll hier nicht näher erläutert werden.

Wie in der Aussagenlogik gibt es auch in der Prädikatenlogik eine Interpretation, die jedoch sehr viel reichhaltiger gestaltet ist (Semantik der Prädikatenlogik). Dabei ist der grundlegende semantische Begriff das **Modell**. Ein Modell der Prädikatenlogik ist ein Paar $\mathcal{M} = (\mathcal{D}, \mathcal{F})$, wobei \mathcal{D} eine Menge und \mathcal{F} eine Funktion ist, die die Elemente aus \mathcal{D} auf Individuenvariablen und die Prädikate auf Relationen abbildet. Die Menge \mathcal{D} wird als **Domäne** (bzw. Struktur) bezeichnet. Diese wird interpretiert als eine Menge von Individuen (z. B. Zahlen, Vektoren, ...), die für das zu betrachtende Problem relevant sind. Das Bild eines Prädikates p ist diejenige Relation R , für die $(x_1, \dots, x_n) \in R$ genau dann gilt, wenn $p(x_1, \dots, x_n)$ zutrifft.

Der Begriff der Interpretation bezeichnet die Deutung der Ausdrücke der Prädikatenlogik in Bezug auf ein Modell. Für die Interpretation von Individuen werden Belegungen verwendet. Eine **Belegung** g zu einem Modell \mathcal{M} ist eine Abbildung mit der Menge aller Individuenvariablen als Definitionsbereich und \mathcal{D} als Wertebereich. Also bildet g Individualvariablen auf Individuen aus \mathcal{D} ab. Eine Interpretation \mathcal{I} ist dann ein Paar (\mathcal{M}, g) aus dem Modell \mathcal{M} und der Belegung g in \mathcal{M} . Für eine Formel \mathcal{C} und eine Interpretation \mathcal{I} schreibt man $\mathcal{I} \models \mathcal{C}$, falls \mathcal{C} für die Interpretation \mathcal{I} wahr ist. Man sagt die Interpretation die Formel \mathcal{C} gilt unter der Interpretation \mathcal{I} [CEE⁺01, S. 40ff].

Axiom 3.2.2 (Axiomensystem der Prädikatenlogik 1. Stufe)

\mathcal{A} , \mathcal{B} und \mathcal{C} sind prädikatenlogische Formeln. Ein *Allsatz* behauptet, dass für alle Individuen die nachfolgenden Prädikate zutreffen. Ein *Existenzsatz* behauptet die Existenz eines oder mehrerer Individuen mit den nachfolgend genannten Prädikaten.

1. prädikatenlogische Annahmeneinführung:

$$\mathcal{C} \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})$$

2. prädikatenlogische Verkettungsregel:

$$((\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})) \Rightarrow ((\mathcal{A} \Rightarrow \mathcal{B}) \Rightarrow (\mathcal{A} \Rightarrow \mathcal{C})))$$

3. prädikatenlogische Widerspruchsregel:

$$(\neg \mathcal{C} \Rightarrow \neg \mathcal{A}) \Rightarrow (\mathcal{C} \Rightarrow \mathcal{A})$$

4. Partikularisierung: Ein *Allsatz* kann auf ein beliebiges Individuum angewandt werden, sofern dessen Beschreibung keine Variablen enthält,

die durch weitere Quantoren des Aussatzes gebunden werden. D. h. $(\forall x)B(x) \Rightarrow B(t)$, wenn t frei substituierbar für x in B ist.

5. Versetzung des Quantors: Wenn ein Allquantor vor einer Implikation eine Variable bindet, die nur im bedingten Teil vorkommt, darf der Quantor direkt vor diesen Teil verschoben werden. D. h. $\forall x(A(\dots) \Rightarrow C(\dots)) \Rightarrow (A(\dots) \Rightarrow (\forall x)C(\dots))$, wenn x nicht frei in A vorkommt.
6. Generalisierung: Wenn eine Aussage für eine freie Variable mit einer beliebigen Durchlaufung dieser Variable gilt, dann darf diese Variable durch einen Allquantor gebunden werden.
7. Inferenzregeln: modus ponens

(nach [Spi04, S. 133])

In Ebbinghaus [EJF96, S. 149] wird über das Problem bei der Verwendung der Prädikatenlogik 1. Stufe gesprochen.

3.2.3 Axiomatische Mengenlehre

Eine Analyse der mathematischen Sprache erfordert die Betrachtung der Mengenlehre. Sie bildet ein einheitliches Gerüst für zahlreiche Disziplinen.

Nach Cantor wird unter einer Menge (naiver Mengenbegriff) folgendes verstanden:

„Unter einer Menge verstehen wir jede Zusammenfassung M von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens (welche die Elemente von M genannt werden) zu einem Ganzen.“ ([Ebb94, S. 1])

Allerdings führt eine solche Betrachtungsweise zu Widersprüchen (Antinomien), beispielsweise bei der Interpretation des Beispiels vom Barbier in Sevilla.:

„In Sevilla wird ein Mann genau dann vom Barbier von Sevilla rasiert, wenn er sich nicht selbst rasiert. Rasiert sich der Barbier selbst?“ [Bar94, S. 172]

Daher wurde die *axiomatische Mengenlehre* entwickelt, die nur spezielle axiomatisch festgelegte Klassen von mathematischen Objekten (*Mengen*) betrachtet. Nur diese speziellen Klassen sind zulässig und sie werden als Mengen bezeichnet.

Bemerkung 3.2.1 (Bezeichnungen)

- Kleinbuchstaben x, y, \dots bezeichnen Mengen und Element von Mengen.
- Das Symbol $x \in y$ bedeutet: „ x ist ein Element von y “
- Das Symbol $x \notin y$ bedeutet: „ x ist kein Element von y “
- Das Symbol $\{x | E(x)\}$ bedeutet: „Die Menge aller x mit der Eigenschaft $E(x)$, d. h. $E(x)$ ist wahr.“

3.2.3.1 Axiomatische Mengenlehre nach ZFC

Es gibt verschiedene formale Systeme in der Mathematik; auf eine Diskussion, welches das Beste ist, wollen wir uns an dieser Stelle nicht einlassen und wählen das *Zermelo-Fraenkelsche Axiomensystem* (ZFC) mit Auswahlaxiom aus. Die Sprache der Mengenlehre basiert auf der Prädikatenlogik erster Stufe mit Identität (Kapitel 3.2.2). Hier ist das einzige nicht-logische Zeichen das \in -Zeichen (Axiome nach [Jap93]).

Axiom 3.2.3 (Extensionalitätsaxiom)

Zwei Mengen a und b sind genau dann gleich, wenn sie dieselben Elemente enthalten.

$$\forall x (x \in a \Leftrightarrow x \in b) \Rightarrow a = b$$

Das Extensionalitätsaxiom beschreibt die Bedeutung des Gleichheitszeichen und legt implizit fest, dass Mengen eindeutig angegeben werden können. Der Ausdruck $\forall x (x \in a \Rightarrow x \in b)$ wird auch als $a \subset b$ geschrieben und bedeutet „ a ist Teilmenge von b “.

Axiom 3.2.4 (Paarungsmengenaxiom)

Wenn a, b Mengen sind, dann gibt es eine Menge x , die genau a und b als Element enthält.

$$\exists x \forall y (y \in x \Leftrightarrow y = a \vee y = b)$$

x heißt *ungeordnetes Paar* von a und b . Es wird als $x = \{a, b\}$ geschrieben. Aus dem Axiom 3.2.4 kann die Definition des *geordneten Paares* gefolgt werden.

Definition 3.2.7 (Geordnetes Paar)

Nach Kuratowski [Ebb94, S. 54] ist ein geordnetes Paar gegeben durch:

$$(x, y) := \{\{x\}, \{x, y\}\}$$

wobei $(x, y) \neq (y, x)$ mit $x \neq y$ ist. Ein geordnetes n -Tupel x_1, \dots, x_n ist ein geordnetes Paar gegeben durch:

$$(x_1, \dots, x_{n-1}, x_n) := \{\{(x_1, \dots, x_{n-1})\}, \{(x_1, \dots, x_{n-1}), x_n\}\}$$

Axiom 3.2.5 (Vereinigungsmengenaxiom)

Zu jeder Menge a gibt es eine Menge x , die alle Elemente der Elemente von a enthält.

$$\exists x \forall y (y \in x \Leftrightarrow \exists z (z \in a \wedge y \in z))$$

Dieses Axiom beschreibt die *Vereinigung von Mengen*. Das x wird auch als $\bigcup a$ geschrieben. Weiterhin ist $a \cup b = \bigcup \{a, b\}$ und $a' := a \cup \{a, a\}$.

Axiom 3.2.6 (Potenzmengenaxiom)

Zu jeder Menge a gibt es eine Menge x , die alle Teilmengen von a enthält.

$$\exists x \forall y (y \in x \Leftrightarrow \forall z (z \in y \Rightarrow z \in a))$$

Dieses Axiom erlaubt die Konstruktion verschieden mächtiger, (überabzählbar) unendlich großer Mengen.

Axiom 3.2.7 (Leere Menge)

Es existiert eine leere Menge.

$$\exists x \forall y (\neg(y \in x))$$

Die leere Menge wird als \emptyset bezeichnet.

Axiom 3.2.8 (Unendlichkeitsaxiom)

Es gibt eine Menge, die \emptyset enthält und mit jedem y auch y' .

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \Rightarrow y' \in x))$$

Dies garantiert die Existenz einer Menge, die alle natürlichen Zahlen enthält. Dabei ist $0 = \emptyset$, $1 = 0' = \{0\}$, $2 = 1' = \{0, 1\}$, $3 = 2' = \{0, 1, 2\}$ usw.

Axiom 3.2.9 (Aussonderungssaxiom)

Für alle Mengen a gibt es eine Menge x , die aus allen Elementen y von a besteht, die die Formel $A(y)$ erfüllen.

$$\exists x \forall y (y \in x \Leftrightarrow y \in a \wedge A(y))$$

Das Axiom 3.2.9 besagt, dass durch bestimmte Eigenschaften $A(y)$ eine neue Teilmenge x aus a gebildet (ausgesondert) werden kann. Dies wird bezeichnet durch: $\{y \in a \mid A(y)\}$. Das Aussonderungssaxiom ist eigentlich ein Axiomenschema für eine unendliche Anzahl von Axiomen, da es unendlich viele $A(y)$ gibt.

Axiom 3.2.10 (Ersetzungsaxiom)

Für jede Menge a existiert eine Menge x so, dass für jedes Element y von a , für das ein z existiert, dass das Prädikat $A(y, z)$ erfüllt, ein solches z auch in x existiert.

$$\exists x (\forall y \in a (\exists z A(y, z) \Rightarrow \exists z \in x A(y, z)))$$

Das Axiom 3.2.10 garantieren die Möglichkeit, Mengen in anderen Mengen nach einer bestimmten Vorschrift abzubilden.

Axiom 3.2.11 (Fundierungsaxiom)

Wenn es eine Menge x gibt, die $A(x)$ erfüllt, dann existiert eine Menge y , die $A(y)$ erfüllt, aber keines ihrer Elemente z erfüllt $A(z)$.

$$[\exists x A(x)] \Rightarrow [\exists x (A(x) \wedge \forall y \in x (\neg A(y)))]$$

Das Fundierungsaxiom führt bei den Mengen, die sich selbst enthalten, zum Widerspruch und verhindert somit die Existenz solcher Mengen.

Axiom 3.2.12 (Auswahlaxiom)

Zu jeder Menge x von nicht leeren, zueinander disjunkten Mengen gibt es eine Menge, die von jedem Element von x genau ein Element enthält.

Das Axiom 3.2.12 besagt, dass zu jeder Menge von nicht leeren, zueinander disjunkten Mengen eine Auswahlmenge existiert. Jedoch beschreibt das Axiom nicht, wie diese Auswahl aussieht und wird nicht für endliche Mengen benötigt.

Mit den Zermelo-Fraenkel Axiomen wird eine formale Sprache für die Standardmathematik aufgebaut. Die Syntaxregeln sind durch die Prädikatenlogik erster Stufe mit Identität (Kapitel 3.2.2) mit den Symbolen (\forall , \exists , \Rightarrow , \Leftrightarrow , \neg und \in , $=$) und den Mengenlehresymbolen gegeben. Im Folgenden soll der Begriff der Zuordnung mengentheoretisch präzisiert werden.

Definition 3.2.8 (Kartesisches Produkt)

Das kartesische Produkt ist gegeben durch:

$$X \times Y := \{(x, y) \mid x \in X \wedge y \in Y\}$$

Des Weiteren wird eine Teilmengenbeziehung benötigt:

Definition 3.2.9

Die Menge X heißt Teilmenge der Menge Y , wenn für alle Elemente x gilt:

$$x \in X \Rightarrow x \in Y$$

Für beliebige Mengen sind die Eigenschaften der Teilmengenbeziehung:

1. $X \subset X$ (reflexiv)
2. Wenn $X \subset Y$ und $Y \subset Z$, dann $X \subset Z$ (transitiv)
3. Wenn $X \subset Y$ und $Y \subset X$, dann $X = Y$ (antisymmetrisch)

[Sta00, S. 7]

Mit Hilfe des Begriffs des geordneten Paares und der Teilmengenbeziehung kann nun der Begriff der Relation definiert werden.

Definition 3.2.10 (Relation)

Es seien X, Y Mengen. Eine Teilmenge R von $X \times Y$ heißt **Relation** zwischen X und Y .

Definition 3.2.11

Es sei $R \subseteq X \times Y$ eine Relation zwischen X und Y .

1. Der **Definitionsbereich** von R ist die Menge D_R der Elemente von X , die R -Urbilder eines Elementes von Y sind:

$$D_R := \{a \in X \mid \exists b \text{ mit } (a, b) \in R\}$$

2. Der **Bildbereich** (Wertebereich) von R ist die Menge W_R der Elemente von Y , die R -Bild eines Elementes von X sind:

$$W_R := \{b \in Y \mid \exists a \text{ mit } (a, b) \in R\}$$

[Sta00, S. 21]

Definition 3.2.12 (binäre Relation)

R wird *binäre (zweistellige) Relation in X* genannt, wenn $R \subseteq X \times X$ ist. (Bezeichnung: $xRy :\Leftrightarrow (x, y) \in R$) R heißt:

reflexiv: wenn $\forall x \in X$ gilt xRx ;

irreflexiv: wenn für kein $x \in X$ gilt xRx ;

transitiv: wenn $\forall x, y, z \in X$ gilt: wenn xRy und yRz , so xRz ;

antisymmetrisch: wenn $\forall x, y \in X$ gilt: wenn xRy und yRx , so $x = y$;

asymmetrisch: wenn $\forall x, y \in X$ gilt: wenn xRy , so nicht yRx ;

symmetrisch: wenn $\forall x, y \in X$ gilt: wenn xRy , so yRx ;

reflexive Halbordnung: wenn R reflexiv, transitiv und antisymmetrisch;

irreflexive Halbordnung: wenn R irreflexiv, transitiv und asymmetrisch;

Äquivalenzrelation: wenn R reflexiv, transitiv und symmetrisch.

[Sta00, S. 22f]

Ein zentraler Begriff in der Mathematik ist der Begriff der Abbildung (Funktion, Operator):

Definition 3.2.13 (Abbildung)

Es seien X, Y Mengen

1. f heißt **Abbildung von X in Y** , wenn
 - (a) f eine Relation von X in Y ist;
 - (b) $(x, y) \in f \wedge (x, z) \in f \Rightarrow y = z$ mit $x \in X$ und $y, z \in Y$
2. Eine Abbildung f wird **surjektiv** genannt, wenn $W_f = Y$ ist.

3. Eine Abbildung f wird **injektiv** genannt, wenn

$$((x_1, y) \in f \wedge (x_2, y) \in f) \Rightarrow x_1 = x_2 \text{ mit } x_1, x_2 \in X \text{ und } y \in Y.$$

4. Eine Abbildung f heit **bijektiv**, wenn f surjektiv und injektiv ist.

[Sta00, S. 25], [Wue95]

In der Literatur wird die Relation von X in Y auch als *Graph von f* bezeichnet. Dadurch wird zwischen dem Prozess des Abbildens von X nach Y und der mengentheoretischen Bedeutung unterschieden. Dies wird in dieser Arbeit im Folgenden aber nicht getan, d. h. es wird $f = \text{Graph}(f)$ angenommen.

3.3 Sprachstrukturen

*Die Mathematik gehört zu jenen Äußerungen
menschlichen Verstandes, die am wenigsten von
Klima, Sprache oder Traditionen abhängen.
(Ilja Ehrenburg, russ. Autor, 1891-1967)*

Die *Sprache der Mathematik* ist mehr als eine gewöhnliche Fachsprache (Kapitel 2.4.4). Sie ist Vorbild für viele Wissenschaften und somit auch deren Fachsprachen, da sie eine „klare durchgängige Strukturierung ihrer Inhalte/Erkenntnisse und [ein] darauf operierende[s] Begriffs-Beziehungsgeflecht“ [Jes04, S. 94] aufweist. Sie besitzt dadurch eine strenge Systematik aus gewissenhaft hergeleiteten Sätzen, die aus einem endlichen Theoremgebäude aufgebaut werden. Definitionen sind dabei die Strukturierungselemente. Allerdings wird die Mathematisierung in den Wissenschaften oft als Allheilmittel gesehen, was nicht unbedingt deren Verstehensprozess und Genauigkeit fördert.

In diesem Kapitel werden die sprachlichen Strukturen der Mathematik und ihre Besonderheiten herausgearbeitet. Um einen grundlegenden Überblick über eine Domäne der Mathematik zu gewährleisten, sollen Texte aus Lehrbüchern verwendet werden. Lehrbücher, insbesondere für Studierende im Grundstudium, nehmen für sich in Anspruch, eine besonders strenge mathematische Sprache zu benutzen [Fis00], [BF00], [BF96], [Kö01], [Rud02], [Wue95]. Es soll daher hauptsächlich Lehrbuchliteratur aus dem Bereich der linearen Algebra verwendet werden, da diese sehr strukturiert aufgebaut ist. Es werden aber auch Beispiele aus der Analysis vorgestellt.

Auf Grundlage der Strukturierungsmechanismen in der Mathematik ergeben sich für linguistische Betrachtungen der mathematischen Sprache vier Ebenen (Abbildung 3.3), die in den nachfolgenden Kapiteln näher betrachtet werden:

1. Entitätenebene (Pragmatik)
2. Binnenstrukturebene (Diskurssemantik)
3. Satzebene (Syntax und Satzsemantik)
4. Wort- und Symbolebene (Morphologie und lexikalische Semantik)

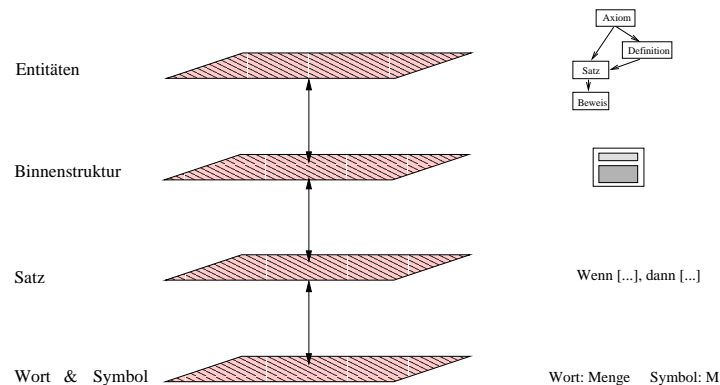


Abbildung 3.3: Darstellungsebenen der mathematischen Sprache

3.3.1 Entitätenebene

Mathematische Theorien werden *deduktiv* konstruiert. Aufbauend auf einer möglichst geringen Anzahl von „undefinierten Begriffen“ (Grundbegriffen) und „unbewiesenen Sätzen“ (Grundsätze⁴) werden weitere Sätze bewiesen [WR84].

„Jede mathematische Theorie ist eine Aneinanderreihung von Sätzen, deren jeder aus den vorhergehenden abgeleitet wird: dies geschieht im Einklang mit den Regeln jenes logischen Systems, das unter dem Namen formale Logik bekannt ist.“ [Bou02, S. 143]

Die Grundsätze werden in der Mathematik auch als **Axiome** bezeichnet. Die Mengenlehre (Kapitel 3.2.3) bildet dabei das axiomatische Fundament der mathematischen Theorien. Auf Grundlage der Axiome der Mengenlehre und der Prädikatenlogik erster Stufe bzw. der elementaren Schlussregeln (Kapitel 3.2.2) werden Aussagen der zu betrachtenden mathematischen Theorie bewiesen.

Durch den dadurch bedingten strukturellen Aufbau wird eine mathematische Theorie in kompakte Einheiten — **Entitäten**⁵ — zerlegt, die untereinander in vorgegebenen Beziehungen stehen [Jes04, S. 93]. Entitäten entsprechen dann

⁴Russell und Whitehead verwendeten die Bezeichnung *Grundsätze* [WR84]. Heutzutage wird der Begriff Axiom verwendet.

⁵Sie werden auch als *Konstrukte* bezeichnet (Kapitel 2.2).

den Definitionen, Theoremen, Sätzen, Korollaren, Propositionen und Lemmata. Hinzu kommen Beweise und Beispiele.

In der Abbildung 3.4 wird eine grobe hierarchische Anordnung der vorhandenen Entitäten dargestellt. Auf der obersten Hierarchieebene (*Axiomenebene*) einer mathematischen Theorie befinden sich die Axiome und Definitionen zu den grundlegendsten Begriffen. Diese beschreiben die Prinzipien einer zu betrachtenden mathematischen Theorie. In der darunterliegenden *Hauptebene* existiert eine stereotype Basisstruktur, bestehend aus Definition, Satz und Korollar. Das zentrale Element ist der Satz, der in verschiedensten Ausprägungen — als Fundamentalsatz, als Theorem oder als „einfacher“ Satz — auftreten kann. Jeder Satz muss aus vorhergehenden Sätzen oder den Axiomen bewiesen werden. Unter der Ebene des Satzes existieren in der Hierarchie die Korollare und Lemmata. Korollare folgen direkt aus einem Satz, Lemmata sind in Beweisen verwendete Hilfssätze. Des Weiteren können auf Sätze und Definitionen Propositionen⁶ folgen, die ebenfalls wieder bewiesen werden⁷. Eine weitere Entitätenstruktur bilden die Beispiele, die häufig in Lehrbüchern zu finden sind. Diese treten hauptsächlich nach Definitionen oder wichtigen Sätzen auf und dienen der anschaulichen Erläuterung der zu betrachtenden mathematischen Aussage.

Bei genauerer Betrachtung der Entitätenstruktur wird deutlich, dass diese Struktur inhärent die pragmatische Struktur widerspiegelt. Sowohl die Entitätenstruktur als auch die verschiedenen Entitäten zeigen eindeutig die Intention an, die ein mathematischer Text vermitteln soll. Die Bestimmung dieser Intentionen ist sicherlich ein wichtiger Schritt für die Analyse der mathematischen Sprache. Problematisch sind mathematische Texte, in denen nicht explizit die Entitäten angegeben werden⁸. Hierbei müssen die Entitäten zuerst aus dem Text ermittelt werden.

⁶In der deutschsprachigen Literatur wird die Proposition auch häufig als *Bemerkung* bezeichnet.

⁷Wird in der deutschsprachigen Literatur die Bezeichnung *Bemerkung* verwendet, so können, aber müssen nicht, Beweise folgen. Wenn kein Beweis angegeben wird, entspricht dies dem in der angelsächsischen Literatur verwendeten Begriff *remark*. Dabei wird meistens ein Hinweis auf einen mathematischen Satz gegeben, dessen Beweis in diesem Zusammenhang nicht wichtig ist und daher ausgelassen werden kann.

⁸Zu solchen Texten gehören z. B. einleitende Textabschnitte zu Sachgebieten.

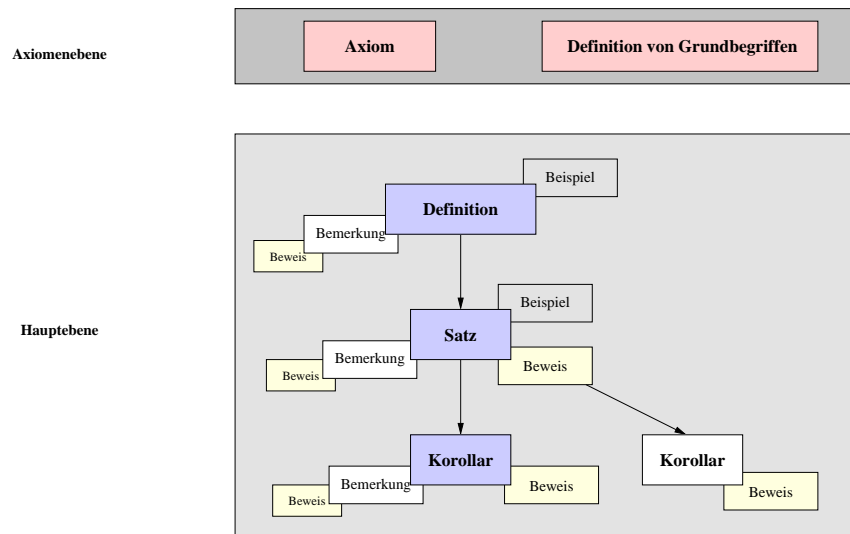


Abbildung 3.4: Hierarchie der Entitäten

In den weiteren Abschnitten sollen nun die Intentionen der einzelnen Entitäten und ihre Charakteristika beschrieben werden.

Axiom. Das Wort *Axiom* stammt aus dem gr.-lat. Wort *axóma* und bedeutet *Grundwahrheit* [Dud01]. Ein Axiom ist eine als wahr angenommene Aussage (bzw. ein als wahr angenommener Satz), die nicht innerhalb der zu betrachteten mathematischen Theorie begründbar ist (Aristoteles). Es wird auch vom *Satz 0. Stufe* gesprochen. Axiome dienen daher als Fundament für deduktiv beweisbare Theorien. Werden in einem Axiom mehrere Aussagen zusammengefasst, wird ein solches Axiom als *Axiomenschema* bezeichnet.

Für die Beschreibung einer mathematischen Theorie werden mehrere Axiome benötigt, die nicht im Widerspruch zueinander stehen dürfen. Ein solches System wird dann als *Axiomensystem* bezeichnet. Beispiele für Axiomensysteme sind die Mengenaxiome, Anordnungs- und Vollständigkeitsaxiome für die reellen Zahlen usw. Axiome bzw. Axiomensysteme treten nur in geringer Anzahl auf.

Definition. Definitionen nehmen in der mathematischen Theorie eine besondere Stellung ein. Ihre Hauptaufgabe ist es, die Übersichtlichkeit und somit

auch die Verständlichkeit einer Theorie zu gewährleisten. Der Begriff *Definition* stammt aus dem lat. *dēfinīre* und bedeutet *abgrenzen* bzw. *bestimmen*, wobei *finis* für *Grenze* steht [Dud01]. Eine Definition dient als Abkürzung von Sachverhalten, die eine wichtige Bedeutung in der zu betrachtenden mathematischen Theorie besitzen, wie z. B. *Vektorraum*, *lineare Abbildung*, *Matrizen*, *Determinante* usw. In der Definition müssen Aussagen genau geklärt und bestimmt werden. Sie werden dabei mit Hilfe von anderen, schon vorher definierten Begriffen und Relationen beschrieben.

Zusätzlich kann den Abkürzungen ein Name, ein Symbol oder eine Symbolfolge zugeordnet werden.

Beispiel

- Die Bezeichnung **Gauss-Verfahren** steht für ein Verfahren, dass eine Matrix in die Zeilenstufenform bringt.
- Der **Vektorraum** kann durch das Symbol **V** beschrieben werden.

Insbesondere hat eine Definition einen Gültigkeitsbereich, der durch den Sachverhalt der mathematischen Theorie geregelt wird.

Es wird zwischen zwei grundlegenden Arten von Definitionen unterschieden:

1. **Explizite Definitionen** werden in der Mathematik sehr häufig verwendet, indem ein Oberbegriff und die zugehörigen Eigenschaften angegeben werden.
2. **Implizite Definitionen** werden häufig in Axiomen verwendet. Es wird ein Begriff durch Angabe von Eigenschaften definiert, ohne ihn anzugeben⁹.

Definitionen dürfen nach Aristoteles keine logischen Widersprüche, Mehrdeutigkeiten und Zirkelschlüsse enthalten. Sie charakterisieren die Merkmale, die nur für den zu definierenden Begriff existieren. Eine Definition besitzt daher keinen Wahrheitswert, jedoch muss der Begriff adäquat beschrieben werden.

⁹Ein Beispiel sind die Axiome von Peano, die die natürlichen Zahlen implizit definieren.

Satz. In der Mathematik sind Sätze bewiesene Aussagen. Es kann zwischen Sätzen 1. Ordnung und Sätzen 2. Ordnung unterschieden werden. Sätze 1. Ordnung werden direkt aus Axiomen hergeleitet. Sätze 2. Ordnung werden aus Sätzen 1. Ordnung abgeleitet. Faktisch wird dies aber nicht gegeneinander abgegrenzt.

Des Weiteren werden drei verschiedene Funktionen von Sätzen differenziert, die durch die prädikatenlogische Struktur induziert werden:

1. **Einzelaussage:** Einzelaussagen sind Aussagen über ein konkretes Objekt. „Die Menge aller rationalen Zahlen ist abzählbar.“ [Rud02, S. 33]
2. **Existenzaussage:** Existenzaussagen werden durch den Existenzquantor induziert und beschreiben die Behauptung über die Existenz eines Objektes mit bestimmten Eigenschaften. „[...] Dann existiert ein $k \in \{1, \dots\}$ so, dass $b_1, \dots, b_{k-1}, a, b_{k+1}, \dots, b_n$ linear unabhängig sind. [...]“ [Wue95, S. 362]
3. **Allaussage:** Allaussagen werden durch den Allquantor induziert und beschreiben die Behauptung, dass eine Eigenschaft für alle Elemente einer Grundmenge gilt. „Alle unendlichen Teilmenge einer abzählbaren Menge sind abzählbar.“ [Rud02, S. 29]

Wichtige Sätze werden häufig mit Namen gekennzeichnet. Dies dient zur besseren Verständlichkeit des zu betrachtenden Sachverhalts. Weniger berühmte Sätze werden schlicht nummeriert wie z. B. „Satz 1“. Diese Bezeichnung ist für Sätze nicht verbindlich und dient nur zur leichteren Orientierung.

In der deutschen Lehrbuchliteratur gibt es verschiedene Typen von Sätzen: Ein **Theorem** ist ein Lehrsatz oder Grundsatz. Es steht für besonders wichtige Sätze, die in der mathematischen Theorie auftreten und deshalb ausgezeichnet werden sollen (z. B.: „*Theorema egregium*“). Ein Theorem, in dem ein ganzes Teilgebiet der Mathematik gipfelt, wird als **Fundamentalsatz** bzw. **Hauptsatz** bezeichnet. Er kommt meistens nur ein- oder zweimal pro Fachgebiet vor (z. B. „*Fundamentalsatz der Algebra*“, „*Hauptsatz der Differential- und Integralrechnung*“). Entsprechend gibt es noch einfache Sätze, die wie oben beschrieben je nach ihrer Wichtigkeit einen Namen erhalten oder namenlos bleiben. Ein wichtiges Kennzeichen von Sätzen ist, dass nach ihnen fast immer die Entität *Beweis* folgt.

Korollar. Der Begriff *Korollar* stammt aus dem lat. *corollar* und bedeutet *Kränzchen*¹⁰. Ein Korollar im mathematischen Sinne ist eine „triviale“ Schlussfolgerung, d. h. eine Sammlung von Folgerungen, die sich aus einem vorhergehenden Satz ohne großen Aufwand ergeben. In diesem Zusammenhang ist auch der **Hilfssatz** zu nennen, der eine rein technische Aussage beschreibt. Dieser wird in der Regel nur im Beweis eines Satzes verwendet und nicht mehr in der weiteren Theorie.

Lemma. Der Begriff *Lemma* stammt ebenfalls aus dem gr.-lat. und bedeutet *Titel* oder *Stichwort* [Dud01]. Ein Lemma ist ein Hilfssatz, der im Rahmen einer Beweisführung erstellt wird, aber auch einen wichtigen Schlüsselgedanken kennzeichnet (z. B.: „*Lemma von Zorn*“, „*Lemma von Sperner*“). Dies unterscheidet es vom Hilfssatz.

Proposition. *Propositionen* stehen im Lat. für Satz [Dud01]. In der deutschen Literatur werden sie auch häufig als *Bemerkung* bezeichnet. Damit sind Aussagen gemeint, die nicht so wichtig wie ein Satz sind, aber auch nicht den Rang eines Hilfssatzes besitzen.

Beweise. In einer deduktiven Theorie wie der Mathematik müssen Aussagen bewiesen werden, wenn diese nicht dem Entitätentyp Axiom oder Definition entsprechen. Eine Definition besitzt keinen Wahrheitswert und ist somit im Sinne der Logik keine Aussage, die bewiesen werden muss.

Der Vorgang des Beweisens bedeutet, aus bekannten Aussagen (Axiomen, Sätzen) mittels Schlussregeln neue Aussagen zu bestätigen. Dabei werden in der Mathematik grundsätzlich zwei verschiedene Beweisarten unterschieden: *direkte* und *indirekte Beweise*. Beim direkten Beweis wird ausgehend von Voraussetzungen (Prämissen) die Behauptung gefolgert. Beim indirekten Beweis (*Reductio ad absurdum*) wird das logische Gegenteil der zu betrachtenden Behauptung als wahr angenommen. Eine solche Behauptung muss dann zu einem Widerspruch zu den Voraussetzungen geführt werden.

Des Weiteren existieren noch einige spezielle Beweisbezeichnungen. Der *Beweis durch vollständige Induktion* ist ein Beweisschema, bei dem die Gültigkeit des

¹⁰Diminutivbildung des Wortes *Korona* [Dud01]

Schemas vorausgesetzt wird. Dabei muss gezeigt werden, dass bestimmte Voraussetzungen (Prämissen) erfüllt werden, woraus in diesem Beweisschema dann automatisch die Behauptung folgt. Existenzsätze werden durch *konstruktive* oder *nicht-konstruktive Beweise* nachgewiesen. Der konstruktive Beweis beschreibt eine Lösungsmethode, die entweder explizit vorgestellt oder skizziert wird. Beim *nicht-konstruktiven* Beweis wird anhand von Eigenschaften auf die Existenz einer Lösung geschlossen. Oft wird dies durch einen indirekten Beweis gezeigt. Bei Äquivalenzaussagen wird häufig eine *Fallunterscheidung* durchgeführt, um zuerst eine Richtung einer Implikation und dann die umgekehrte Richtung zu beweisen.

Eine tiefergehende Betrachtung von Beweisen soll hier nicht diskutiert werden. Zur Vereinfachung der Analyse der mathematischen Sprache sollen Beweise in einem ersten Schritt nicht verwendet werden. Somit sind sie nicht Gegenstand dieser Arbeit. Jedoch wird im Kapitel 5 auf die Wichtigkeit der Beweise in der Mathematik eingegangen.

Beispiel. Ein *Beispiel* steht für einen speziellen bzw. einzelnen Fall, der etwas Allgemeines näher erläutert oder erklärt. Durch den strukturellen Aufbau der Mathematik kommt es in der Lehrbuchliteratur sehr häufig vor, dass Beispiele im Textgefüge ebenfalls eine Entität darstellen und somit eindeutig zu unterscheiden sind. Es gibt noch eine weitere Art von Beispielen die *Gegenbeispiele*. Dies sind Beispiele, die eine Definition oder die Voraussetzung eines Satzes nur fast erfüllen. Auch dieser Typ der Entität soll zunächst nicht weiter betrachtet werden.

3.3.2 Binnenstrukturebene

Mathematische Texte besitzen durch die Existenz der oben beschriebenen Entitäten einen strukturierten Textaufbau, der den gesamten Text in Blöcke zerlegt. Aber auch die Entitäten selbst weisen eine interne Struktur auf, die für die einzelnen Typen von Entitäten charakteristisch sein können. Diese Struktur wird als *Binnenstruktur* bezeichnet [Jes04]. Alle Sätze sowie Definitionen besitzen die grundlegende Struktur (Abbildung 3.5): **Voraussetzungen, Aussagen und Eigenschaften**¹¹.

¹¹Eigenschaften können wieder Aussagen enthalten.

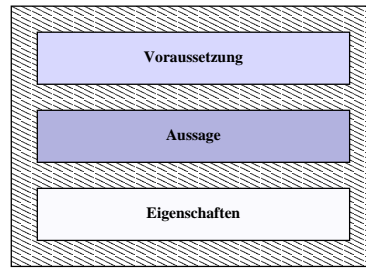


Abbildung 3.5: Binnenstruktur von Definitionen und Sätzen

Die einzelnen Elemente der Binnenstruktur sind grammatikalisch an ihren Verwendungszweck angepasst (Kapitel 3.3.3). In *Voraussetzungen* werden diejenigen mathematischen Objekte festgelegt, die für die Betrachtung der Entität von Bedeutung sind. So werden z.B. Symbole oder Wertebereiche festgelegt. Die syntaktischen Strukturen in den *Aussagen* sind an die entsprechenden Entitätentypen (Kapitel 3.3.1) gebunden. In **Definitionen** wird eine Aussage in *Definiendum* und *Definiens* zerlegt. Das Definiendum ist eine Bezeichnung für das neu definierte mathematische Objekt oder den neu definierten mathematischen Sachverhalt. Das Definiens ist derjenige Ausdruck, der das Definiendum mithilfe der mathematischen Sprache beschreibt. Eine Definition kann weiterhin noch Eigenschaften auflisten, die das Definiendum besitzt (Abbildung 3.6). Dabei können Eigenschaften selbst wieder Eigenschaftslisten enthalten.

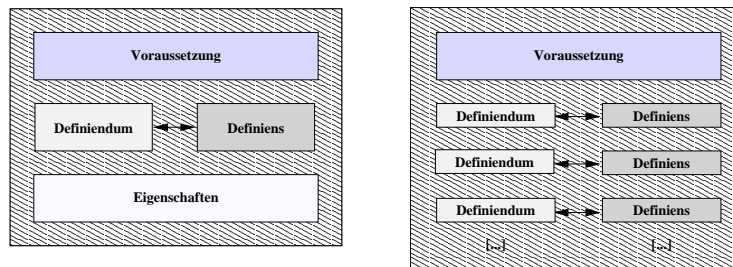


Abbildung 3.6: Binnenstruktur der Definition [Jes04, S. 120]

Beispiel 3.3.1 (Definition)

In der folgenden Definition sind die verschiedenen Binnenstrukturen ebenfalls

farbig unterlegt. Grün steht für die Voraussetzungen, Blau für das Definiens und Rot für das Definiendum. Eine Eigenschaftsaufzählung zeigt dieses Beispiel nicht.

Definition

Gegeben sei eine Folge $\{p_n\}$. Man betrachte eine Folge $\{n_k\}$ positiver Zahlen mit $n_1 < n_2 < \dots$. Dann heißt die Folge $\{p_{n_k}\}$ **eine Teilfolge von $\{p_n\}$** . [Rud02, S. 59]

Die **Satzbinnenstruktur** kann nicht so einfach strukturiert werden wie die Definitionsbinnenstruktur. In Kapitel 3.3.3 wird gezeigt, dass verschiedene Entitätentypen, wie z. B. Satz und Korollar, nicht in ihren syntaktischen Strukturen zu unterscheiden sind. Als gemeinsames Merkmal besitzen sie die syntaktische Struktur der Prädikatenlogik (Kapitel 3.2.2). Die Aussagen werden daher mithilfe der sprachlichen Äquivalente der logischen Operatoren aufgebaut. Verwendung finden hierbei die Implikation (Abbildung 3.7) und die Äquivalenz (Abbildung 3.8).

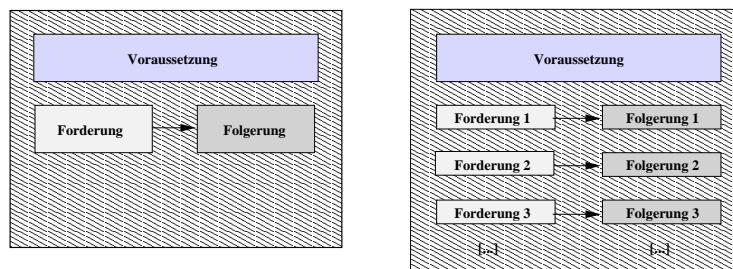


Abbildung 3.7: Binnenstruktur des Satzes für Implikationen [Jes04, S. 121]

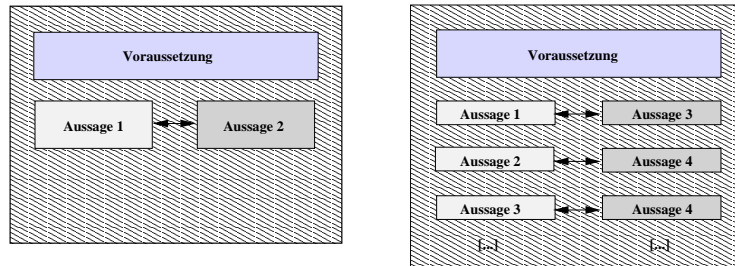
Beispiel 3.3.2 (Satz mit Implikation)

Im folgenden Satz sind die verschiedenen Binnenstrukturen farbig unterlegt. Grün steht für die Voraussetzung¹², Blau für die Forderung und Rot für die Folgerung.

¹²Diese kennzeichnet allgemeine Voraussetzungen, die für den Satz gemacht werden. Meistens werden diese einfach weggelassen.

Satz

Sei \mathcal{A} eine Algebra reeller stetiger Funktionen auf einer kompakten Menge K . Separiert \mathcal{A} die Punkte auf K und verschwindet in keinem Punkt von K , dann besteht die gleichmäßig abgeschlossene Hülle \mathcal{B} von \mathcal{A} aus allen stetigen Funktionen auf K . [Rud02, S. 189]



Abbildungung 3.8: Binnenstruktur des Satzes für Äquivalenzen [Jes04, S. 121]

Beispiel 3.3.3 (Satz mit Äquivalenz)

In dem folgenden Satz sind die verschiedenen Binnenstrukturen wieder farbig unterlegt. Grün Kennzeichnung steht für die Voraussetzung¹³, Blau für die Aussage 1 und Dunkelgrau für die Aussage 2.

Satz

Für $F \in \text{End}(V)$ sind folgende Bedingungen äquivalent:

1. F ist surjektiv.
2. $\det F \neq 0$

[Fis00, S. 203]

Eine weitere einfache Satzinnenstruktur ist die *einfache Aussage*. Außer den Voraussetzungen besitzt die einfache Aussage keine weiteren inneren Strukturierungen (Abbildung 3.9).

¹³Diese kennzeichnet allgemeine Voraussetzungen, die für den Satz gemacht werden. Meistens werden diese einfach weggelassen.

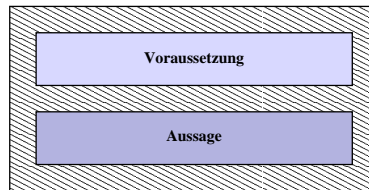


Abbildung 3.9: Binnenstruktur des einfachen Satzes

Beispiel 3.3.4 (Einfache Aussage)

Die Teilfolgengrenzwerte einer Folge $\{p_n\}$ in einem metrischen Raum X bilden eine abgeschlossene Teilmenge von X . [Rud02, S. 60]

Axiome zeigen eine eigenwillige Struktur. Sie sind zum Teil mit den Sätzen verwandt, zeigen aber auch definitionscharakteristische Strukturierungen. Außerdem müssen Axiome zwangsläufig mit außermathematischen Begriffen operieren. Daher lässt sich bei ihnen keine klare Binnenstruktur erkennen¹⁴.

Zwischen den einzelnen Binnenstrukturen existieren Abhängigkeiten. So werden in den Voraussetzungen Symbole und Bezeichnungen festgelegt, die für eine Interpretation der Aussage wichtig sind. Die Identifikation solcher Zusammenhänge wird dann durch die Diskurssemantik beschrieben (Kapitel 2.4).

3.3.3 Satzstrukturebene

Die Satzstrukturen in mathematischen Definitionen, Sätzen und Axiomen zeigen stereotype syntaktische Konstruktionen wie z. B. Metaphern und Phrasen. So sind Definitionen an ihrer syntaktischen Struktur und durch die Verwendung von Schlüsselverben wie z. B. *heißen* leicht zu erkennen. Sätze dagegen sind zwar als solche wahrzunehmen, es existieren aber keine Unterscheidungsmerkmale zwischen den einzelnen Satztypen (*Satz*, *Korollar* und *Proposition*).

In den hier betrachteten Texten können mathematische Aussagen stets als wahr angenommen werden. Dementsprechend entfällt der Teil der pragmatischen Analyse (Kapitel 2.4), welche der beabsichtigte Wirkung eines Satzes bezüglich eines Kommunikationspartners betrachtet. Die sprachlichen Konstruktionsmöglichkeiten sind somit begrenzt.

¹⁴Beispiele für Axiome im Kapitel 3.2.3

Mathematische Aussagesätze werden durch die Prädikatenlogik 1. Stufe charakterisiert. So werden mit Hilfe einer geringen Anzahl von verwendeten logischen Operatoren (Junktoren) und Quantoren (Kapitel 3.2.2) syntaktisch einfache Sätze konstruiert. Für diese logischen Operatoren und Quantoren steht nur eine begrenzte Anzahl an syntaktischen Konstruktionen zur Verfügung. Die dabei entstehenden Satzstrukturen sind meistens kurz, klar und leicht überschaubar konstruiert. Durch die Verwendung von feststehenden Phrasen werden somit viele syntaktische und semantische Ambiguitäten vermieden, weil die einzelnen Konstruktionen entsprechend festgelegte Bedeutungen besitzen. Somit wird eine syntaktische und semantische Analyse wesentlich vereinfacht. Ein weiterer Hinweis für die Möglichkeit einer strafferen syntaktischen Analyse ist die Verwendung nur einer Zeitform — des *Präsens* — und die Verwendung eines bestimmten Modus — entweder Indikativ oder Konjunktiv — wobei der Konjunktiv in Voraussetzungen verwendet wird.

Der Genus Verbi — Passiv und Aktiv — wird in der deutschen Gegenwartssprache ungleich verwendet: 93% entfallen auf den Aktiv und 7% auf den Passiv [Dud01, S. 172]. In der Mathematik dagegen tritt der Passiv höchst selten auf. So wird in Definition selten die Bezeichnung „*wird genannt*“ verwendet.

Beispiel

[...] *Dann wird der Ausdruck $\sum_{k=c}^{\infty} a_k$ Reihe genannt.* [Wue95, S. 119]

Außerdem scheint die Anzahl der verwendeten Verben gering zu sein. Dabei treten besonders häufig die Wörter „*sein*“ sowie „*heißen*“, „*existieren*“, „*geben*“, „*folgen*“ usw. auf. Einige dieser Wörter stehen als Äquivalent für logische Operatoren („*folgen*“) und Quantoren („*existieren*“). Andere Wörter hingegen sind an die Entitäten gebunden. So findet man in [Fis00] bei ca. 34 explizit als Entität angegebenen Definitionen 28 mal das Wort *heißen* (82 %). In [Wue95] wird *heißen* bei 173 Definitionen 92 mal (53 %) gebraucht und „ \Leftrightarrow “ 31 mal (15 %).

Durch die Verwendung von mathematischen Symbolen entstehen grundlegende Schwierigkeiten für die syntaktische und semantische Analyse. Mathematische Symbolfolgen wie z.B. Formeln weisen eine ihnen eigene Syntax auf, die zwar prinzipiell einfach ist und auch durch die Prädikatenlogik strukturiert wird,

jedoch nicht kompatibel zur syntaktischen Struktur der natürlichsprachlichen Texte ist.

Mathematische Zeichen werden häufig im Zusammenhang mit und als Nominalphrasen verwendet und stellen in mathematischen Texten eine äquivalente Bezeichnung für diese dar.

Beispiel

1. *Jedes Polynom $f \in \mathbb{C}(f)$ zerfällt in Linearfaktoren, [...]* [Fis00, S. 63]
2. *\mathbb{R} zusammen mit der Addition $+$ ist eine abelsche Gruppe.* [Fis00, S. 54]

Vielfach werden im weiteren Verlauf eines Textes nur noch diese Abkürzungen verwendet, so dass mathematische Zeichen in einer linguistischen Analyse berücksichtigt werden müssen. Die interne Struktur solcher Symbolfolgen kann komplex sein. So kann es passieren, dass in ihnen weitere für die linguistische Analyse wertvolle Informationen enthalten sind.

Beispiel

$\{a_n\}_{n \in \mathbb{N}}$ beschränkt \Leftrightarrow es gibt ein $c \in \mathbb{R}$ mit $|a_n| \geq c$ ($n \in \mathbb{N}$).
[Wue95, S. 108]

Des Weiteren enthalten mathematische Entitäten auch grammatikalisch nicht korrekte Strukturen, z. B. „ f injektiv.“. So fehlt häufig bei Aufzählungen das entsprechende Verb. Solche Strukturen gilt es zu erkennen und gegebenenfalls zu behandeln.

Eine besondere Problematik entsteht durch die zahlreichen Unterschiede der Sprachstile der verschiedenen Autoren. So verwendet jeder Autor eine für ihn spezifische Sammlung von Phrasen.

Ein mathematischer Text ist aber kein Schulaufsatz, bei dem die Ausdrucksvielfalt die entscheidende Rolle spielt; dennoch steht dem sensiblen Schreiber zur Formulierung eines logischen Schlusses eine Vielfalt von Ausdrucksmöglichkeiten offen. [Beu95, S. 33]

Basiskonstruktionen.

In der mathematischen Sprache treten immer wiederkehrende Satzkonstruktionen auf, die als syntaktische Grundlage die Prädikatenlogik widerspiegeln. Hierbei werden den logischen Operatoren sprachliche Pendants zugeordnet. Sicherlich gibt es dabei eine Vielzahl an sprachlichen Ausdrucksmöglichkeiten für die Formulierung der logischen Operatoren, die insbesondere in den Zwischentexten wie z. B. in Motivationen in der mathematischen Literatur häufig verwendet werden. In den festgelegten Entitäten werden die Zuordnungen dagegen streng gehandhabt, so dass nur eine geringe Anzahl von Standardformulierungen existiert.

Implikationen $A \Rightarrow B$. In Kapitel 3.3.2 wurde festgestellt, dass eine Implikation stets aus einer *Forderung* und einer *Folgerung* besteht. Da Implikationen in mathematischen Texten vielfach verwendet werden, gibt es zahlreiche Standardformulierungen für diesen logischen Operator. Eine der häufigsten Standardkonstruktionen, die in Sätzen verwendet wird, ist:

[VERB1] A , (so/dann) [VERB2] B .

Die Bezeichnung [VERB] steht für ein beliebiges Verb, das in der mathematischen Sprache an dieser Stelle gesetzt wird. Dabei tritt das Verb „sein“ besonders häufig in beiden möglichen Positionen [VERB1] und [VERB2] auf. Dagegen wird das Verb „gelten“ häufig in der Position [VERB2] eingesetzt.

Beispiel 3.3.5 (Implikation I)

- Sind $f, g \in K[t]$ und ist $g \neq 0$, so gibt es dazu eindeutig bestimmte Polynome $q, r \in K[t]$ derart, dass $f = q \cdot g + r$ und $\deg r < \deg g$. [Lor96, S. 15]

Der Satz besteht nach Abbildung 3.10 aus einer Folgerung und einer Forderung. Jede dieser Binnenstrukturen besitzt einen Indikator, der sie als Folgerung oder Forderung kennzeichnet, und einen mathematischen Inhalt.

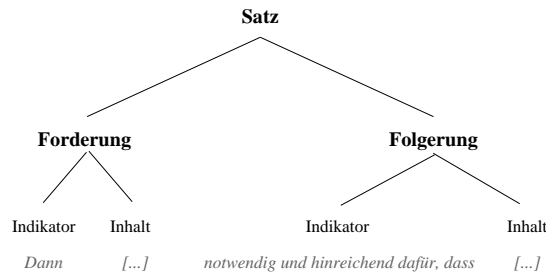


Abbildung 3.10: Basisstrukturanalyse für Implikationskonstruktionen

- *Konvergiert* die Potenzreihe P in einem Punkt $z_0 \neq 0$, *so konvergiert* sie absolut in jedem Punkt $z \in \mathbb{C}$ mit $|z| < |z_0|$. [Kö01, S. 51]

Weitere Konstruktionen, die der obigen Standardformulierung sehr ähnlich sind, aber nicht so oft verwendet werden, sind:

- wenn A [SEIN/GELTEN], dann [SEIN/GELTEN] B ;
- wenn A [GELTEN], so [GELTEN] B ;
- falls A [VERB], dann [VERB] B ;
- B [VERB] (nur/höchstens) dann, wenn A ;

Beispiel 3.3.6 (Implikation II)

Sei $A \subset V$. *Falls* für ein $a \in A$ $T_a(A)$ Teilraum von V , *dann* ist $T_b(A)$ Teilraum für alle $b \in A$, also A ein affiner Teilraum von V . [Wue95, S. 388 (2) – Satz]

Durch die Verwendung der Wörter *hinreichend* und *notwendig* entstehen folgende Bedeutungen: *Hinreichend* steht dafür, dass die Forderung A ausreicht damit auch die Folgerung B wahr ist. Die Bezeichnung *notwendig* steht dafür, dass ohne die Folgerung B die Forderung A nicht erfüllt sein kann¹⁵.

¹⁵ A hinreichend für B : $A \Rightarrow B$; A notwendig für B : $B \Rightarrow A$

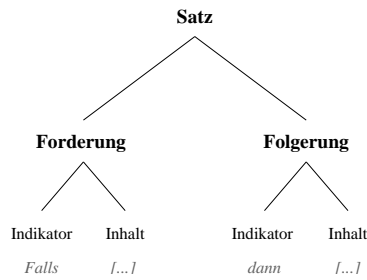


Abbildung 3.11: Basisstrukturanalyse für Implikationskonstruktionen

- A ist hinreichend für/ dies ist hinreichend für/dies ist eine hinreichende Bedingung für B ;
- A ist notwendig für/ eine notwendige Bedingung dafür ist B ;

Beispiel 3.3.7 (Implikation III)

Die offene Menge $U \subset \mathbb{R}^n$ sei wegzusammenhängend und $f : U \longrightarrow \mathbb{R}$ sei differenzierbar.

Dann ist die Bedingung

$$f'(\mathbf{x}, \mathbf{h}) = 0 \text{ für alle } \mathbf{x} \in U \text{ und für alle } \mathbf{h} \in \mathbb{R}^n$$

notwendig und hinreichend dafür, dass f konstant ist. [BF96, S.116]

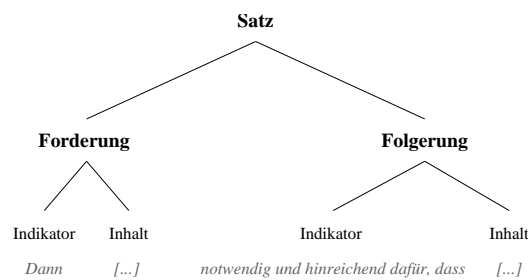


Abbildung 3.12: Basisstrukturanalyse für Implikationskonstruktionen

In der deutschen Lehrbuchliteratur wird auch oftmals das Wort *folgern* anstelle von *implizieren* verwendet. Aus diesen beiden Verben können verschiedene Konstruktionen gebildet werden:

- aus A folgt B
- A dies hat zu Folge, dass/ man kann folgern, dass B
- A folglich [VERB] B ;
- A , dies impliziert B ;
- A , daraus ergibt sich/ daraus erhalten wir/ das bedeutet B ;

Beispiel 3.3.8 (Implikation IV)

Sei V ein K -Vektorraum. Eine endliche Familie (v_1, \dots, v_r) von Vektoren aus V heißt *linear unabhängig, falls gilt*: Sind $\lambda_1, \dots, \lambda_r \in K$ und ist

$$\lambda_1 v_1 + \dots + \lambda_r v_r = 0,$$

so folgt

$$\lambda_1 = \dots = \lambda_r = 0.$$

[Fis00, S.77]

Äquivalenz: $A \Leftrightarrow B$. Im Kapitel 3.3.2 wurde neben der Implikation auch die Äquivalenz als weiterer wichtiger logischer Operator für die Konstruktion von mathematischen Sätzen verwendet. Dabei wird eine Aussage zu einer anderen Aussage äquivalent gesetzt. Auch für die Äquivalenz gibt es Standardkonstruktionen:

- A ist äquivalent zu/ gleichbedeutend mit B ;
- A [gelten] genau dann, wenn B [gelten];
- A [gelten] dann und nur dann, wenn B [gelten];
- A [sein] hinreichend und notwendig für B ;

Beispiel 3.3.9 (Äquivalenz)

- Eine Reihe $\sum_k a_k$ mit Gliedern $a_k \geq 0$ *konvergiert genau dann, wenn* die Folge ihrer Partialsummen beschränkt ist. [Kö01, S.61 – Satz]

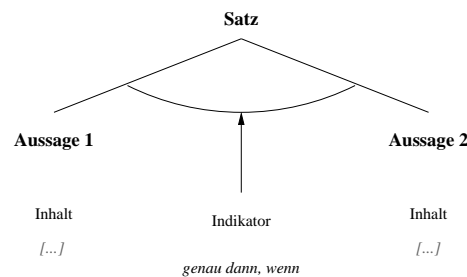


Abbildung 3.13: Basisstrukturanalyse für Äquivalenzkonstruktionen

In der Abbildung 3.13 ist die Struktur des Satzes schematisch dargestellt. Die beiden Aussagen werden durch den Indikator „*genau dann, wenn*“ verbunden.

- Ist X eine endliche Menge, so sind für die Abbildung $f : X \longrightarrow X$ folgende Bedingungen *äquivalent*:

- i) f ist injektiv.
- ii) f ist surjektiv.
- iii) f ist bijektiv.

[Fis00, S.34]

In der Abbildung 3.14 werden drei Aussagen über einen Indikator verbunden.

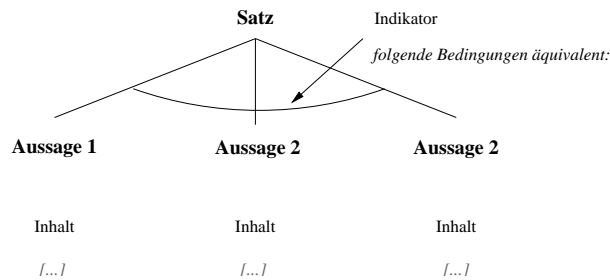


Abbildung 3.14: Basisstrukturanalyse für Äquivalenzkonstruktionen

Negation \neg , Konjunktion \wedge und Disjunktion \vee . Diese drei Junktoren werden im allgemeinen sehr einfach konstruiert. Für die Negation stehen sprachliche Konstrukte wie „*kein*“ und „*nicht*“ zur Verfügung. Für die Konjunktion wird sehr häufig das Wort „*und*“ verwendet. Man kann aber auch Konstruktionen wie „*sowohl A als auch B*“ vorfinden. Für die Disjunktion findet man hauptsächlich „*oder*“-Konstruktionen¹⁶.

Beispiel 3.3.10 (Negation und Konjunktion)

- Sei $f \in \mathbb{R}[t]$ *und* $\lambda \in \mathbb{C}$ eine *nicht* reelle Nullstelle von f , *sowie* $g := (t - \lambda)(t - \bar{\lambda}) \in \mathbb{C}[t]$. [...] [Fis00, S. 64]
- Ist K ein Körper, so ist $\text{char}(K)$ *entweder* Null *oder* eine Primzahl¹⁷. [Fis00, S. 57]

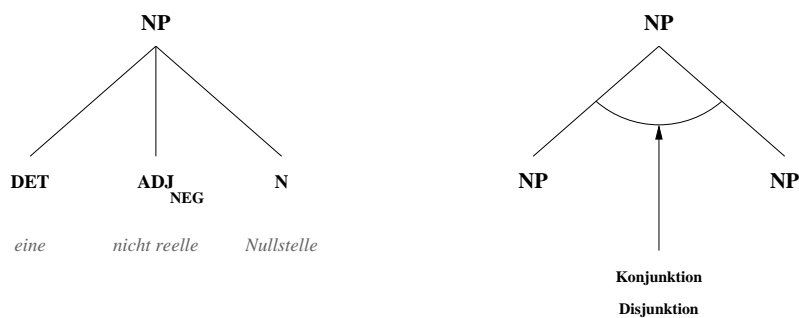


Abbildung 3.15: Basisstrukturanalyse von Quantorenkonstruktionen

¹⁶Allerdings muss mit dem Wort *oder* vorsichtig umgegangen werden, so ist z. B. *entweder* [...] *oder* keine Disjunktion

¹⁷Hier wird implizit sogar mehr gesagt: *Null ist keine Primzahl*.

In Abbildung 3.15 wird auf der linken Seite die Nominalphrase „*eine nicht reelle Nullstelle*“ in einen Artikel, ein Substantiv und ein getyptes Adjektiv zerlegt. Der Typ *NEG* bedeutet, dass das Adjektiv verneint wird. Auf der rechten Seite setzt sich eine Nominalphrase aus zwei weiteren Nominalphrasen zusammen, die durch eine Konjunktion „*und*“ oder eine Disjunktion „*oder*“ verbunden sind.

Quantoren Durch die Verwendung von Quantoren der Prädikatenlogik wird die syntaktische sowie die semantische Struktur reichhaltiger. Wir unterscheiden zwei Arten von Quantoren, Allquantor und Existenzquantor, die ebenfalls in ihrer sprachlichen Form Standardkonstruktionen besitzen. Folgende Standardkonstruktionen werden für den Allquantor verwendet:

- Für alle/jedes/ein beliebiges x [...]
- Jedes / zu jedem x [...]
- Alle x [...]
- Sei x beliebig [...]

Beispiel 3.3.11 (Allquantor)

Für alle Regelfunktionen f_1, f_2, f und [...] [BF00, S.356]

Ein weiterer Quantor ist der Existenzquantor, der nur die Existenz eines bestimmten Objektes kennzeichnet. Auch dafür existieren Standardkonstruktionen:

- Es gibt ein x [...];
- Für ein geeignetes x gilt [...];
- [SEIN/HABEN] ein x [...];

Beispiel 3.3.12 (Existenzquantor)

Jedes Polynom $f \in \mathbb{C}[t]$ zerfällt in Linearfaktoren, d. h. es gibt ein a und $\lambda_1, \dots, \lambda_n \in \mathbb{C}$ mit $n = \deg f$, so dass $f = a(t - \lambda_1) \cdot \dots \cdot (t - \lambda_n)$. [Fis00, S.63]

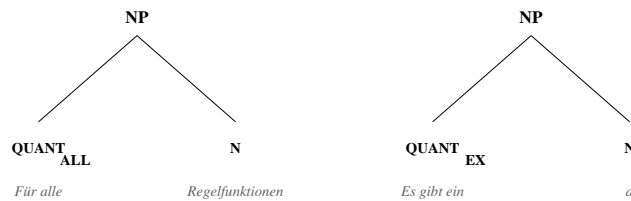


Abbildung 3.16: Basisstrukturenanalyse

In Abbildung 3.16 wird die Nominalphrase in einen Quantor (*QUANT*) und ein Substantiv zerlegt. Der Allquantor wird durch *ALL* und der Existenzquantor durch *EX* getypt.

Mengentheoretische Bezeichnungen. Auch für das nicht-logische Zeichen \in gibt es sprachliche Standardkonstruktionen. Man beachte jedoch, dass in Texten fast ausschließlich die Abkürzung verwendet wird:

1. [...] ist Element von [...]
2. [...] kommt in [...] vor

Beispiel 3.3.13 (Elemente von Mengen)

Für alle Regelfunktionen f_1, f_2, f und alle $c \in \mathcal{R}$ gilt [...]. ([BF00, S.356])

Voraussetzung. Jeder Satz und jede Definition in der Mathematik besitzt in seiner Binnenstruktur die Festlegung von Voraussetzungen, d. h. die Festlegung derjenigen mathematischen Objekten, die in der Entität verwendet werden. Die Aufzählung von Voraussetzungen erkennt man deutlich an der syntaktischen Struktur, die im untenstehenden Kasten schematisch dargestellt wird. Dabei

wird sehr häufig das Verb „sein“ mit den Verbformen „sei“ und „seien“ verwendet.

(es) sei [...], ist [...], für [...], gegeben (ist/sei) [...], es gelte [...]

Beispiel 3.3.14 (Voraussetzung)

1. „**Sei** V ein Vektorraum über den Körper K und T eine linear unabhängige Teilmenge von V .“ [Lor96]
2. „**Ist** F ein Zwischenkörper der algebraischen Erweiterung E/K , so gilt $[E : K]_s = [E : F]_s \cdot [F : K]_s$ “ [Lor96, S.87]
3. „**Für** teilerfremde Ideale I_1, I_2 von R gilt $I_1 I_2 = I_1 \cap I_2$ “ [Lor96, S.54]
4. „**Gegeben** **seien** Abbildungen $f_i : X_i \longrightarrow Y_i, i \in I, X_i \neq \emptyset$, zwischen topologischen Räumen X_i und Y_i . Die Abbildung $f : \prod_{i \in I} X_i \rightarrow \prod_{i \in I} Y_i$ $(x_i)_{i \in I} \mapsto (f_i(x_i))_{i \in I}$ ist genau dann stetig, wenn für alle $i \in I$ die Abbildungen f_i stetig sind.“ [vQ01, S.41]
5. „**Es** **gelte** $f_k \in \mathcal{L}^1(\mathbb{R}^n)$ und ...“ [BF96, S.302]

In Abbildung 3.17 wird die Analyse des einfachen Satzes „Sei V ein Vektorraum über K “ aus Beispiel 3.3.14 skizziert. Das Verb „sei“ wird auf die Position 2 hinter der Nominalphrase verschoben und in den Indikativ umgewandelt¹⁸. Danach wird der Satz nach bekanntem Muster zerlegt (Kapitel 2.4). In einem nächsten Schritt werden die Nominalphrase als das Definiendum und das in der Verbalphrase befindliche Definiens erkannt.

Satzstrukturen in Entitäten

Axiom. An der Satzstruktur kann man Axiome nicht unbedingt erkennen, da sie syntaktisch mit Sätzen und Definitionen gleichzusetzen sind. Daher findet man beide syntaktischen Strukturen wieder, die miteinander vermischt werden.

¹⁸Im mArachna-Projekt werden Prepositionalphrasen (PP) ohne Ausnahme als spezielle Nominalphrase verarbeitet, jedoch zur Zeit noch nicht gesondert aufgeführt.

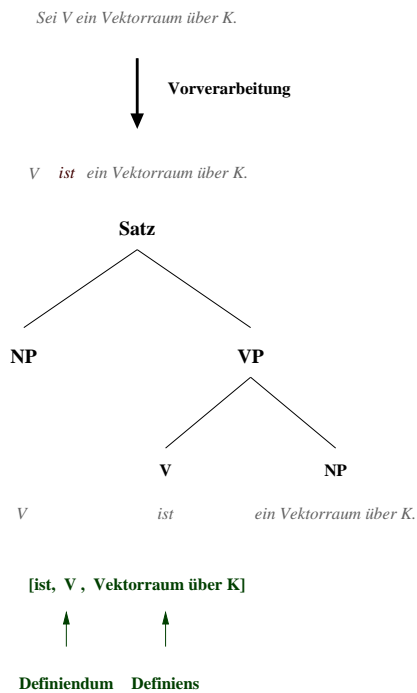


Abbildung 3.17: Analyse der Voraussetzungen

Beispiel 3.3.15 (Axiom 3.2.7)

Leere Menge: Es gibt eine leere Menge.

Definition. Definitionen sind relativ einfache Strukturen, die leicht über das verwendete Verb erkannt werden. Außerdem steht nur eine begrenzte Anzahl von Verben zur Verfügung: „*heißen*“, „*bezeichnen*“, „*nennen*“, „*definieren*“, usw. Häufig ist das Definiendum gegenüber den anderen Wörtern in der Entität hervorgehoben. Ebenfalls sehr häufig werden Klammern verwendet, die eine Nominalphrase einschließen und somit auch ein Definiendum sein können. Diese Art des Definierens wird häufig in der Eigenschaftsaufzählung verwendet, die dann eine Formel bzw. eine Aussage bezeichnet (Kapitel 3.3.4).

Eine häufige Grundstruktur ist:

Definiendum Symbol heißt Definiendum Bezeichnung wenn gilt:

- | |
|---|
| <ul style="list-style-type: none"> • Eigenschaften |
|---|

In den Eigenschaften werden mathematische Aussagen aufgelistet, die ihrerseits auch wieder einen Namen bekommen können. Sie können aber auch nur mathematische Formeln bzw. Aussagen sein.

Beispiel 3.3.16 (Gruppe)

Eine Menge G zusammen mit einer Verknüpfung $*$ heißt Gruppe, wenn folgende [Eigenschaften] erfüllt sind:

G1 $(a * b) * c = a * (b * c)$ für alle $a, b, c \in G$ (Assoziativgesetz)

G2 Es gibt ein $e \in G$ (neutrales Element genannt) mit den folgenden Eigenschaften:

- $e * a = a$ für alle $a \in G$
- Zu jedem $a \in G$ gibt ein $a' \in G$ (inverses Element von a genannt) mit $a' * a = e$.

[Fis00, S.41]

Analyse des Beispiel 3.3.16 Die Definition aus dem Beispiel 3.3.16 kann in Unterabschnitte zerlegt werden:

Teil 1: Eine Menge G zusammen mit einer Verknüpfung $*$ heißt Gruppe, wenn folgende Eigenschaften erfüllt sind.;

Teil 2: $(a * b) * c = a * (b * c)$ für alle $a, b, c \in G$ (Assoziativgesetz);

Teil 3: Es gibt ein $e \in G$ (neutrales Element genannt) mit den folgenden Eigenschaften:

Teil 4: $e * a = a$ für alle $a \in G$

Teil 5: Zu jedem $a \in G$ gibt ein $a' \in G$ inverses Element von a genannt) mit $a' * a = e$.

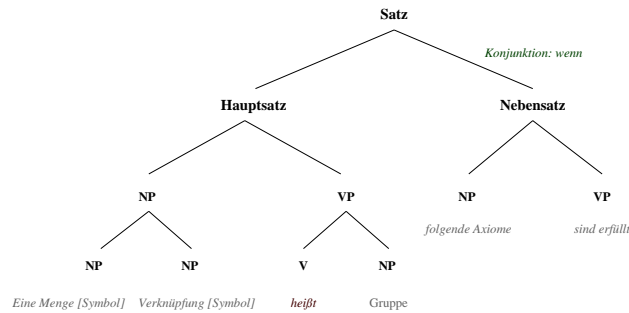


Abbildung 3.18: Syntaktische Analyse des Beispiels 3.3.16 — Teil 1

Teil 1 in der Definition lässt sich in Hauptsatz und Nebensatz zerlegen. Der *Hauptsatz* definiert die *Gruppe*, gekennzeichnet durch das Verb *heißen*. In Abbildung 3.18 ist die syntaktische Analyse des Hauptsatzes auf der linken Seite zu sehen. Die Struktur lässt sich leicht auflösen. Der *Nebensatz* leitet die Eigenschaftsliste ein, gleichzeitig wird durch die Konstruktion „*wenn* [...] *erfüllt sind*“ gesagt, dass die Definition nur gilt, wenn die folgenden Eigenschaften erfüllt werden.

Teil 2, 3 und 4 ist syntaktisch ebenfalls leicht zu analysieren. Eine Formel gefolgt von einer Klammer deutet auf eine Definition hin. Daher wird der Satz durch eine Vorverarbeitungsregel¹⁹ in eine Aussage und eine Definition zerlegt. Diese werden dann getrennt syntaktisch analysiert. Des Weiteren wird hierbei eine Formel $[F1]$ wie ein Substantiv behandelt. Da es kein Substantiv ist, wird im weiteren Verlauf eine Formel einen neuen Worttyp *MathWord* zugeordnet.

Eine weitere Konstruktionsmöglichkeit wird durch mathematische Symbole gegeben. So kann der Äquivalenzoperator „ \Leftrightarrow “ verwendet werden, um ein Definiendum zu definieren, wobei es dann leicht als solches zu erkennen ist wegen „:“:

Beispiel 3.3.17

Sei $A \subset \mathbb{R}$

- (1) A *offen* \Leftrightarrow zu $x \in A$ existiert eine ε -Umgebung $U_\varepsilon(x)$ von x mit $U_\varepsilon(x) \subset A$.
[...]

¹⁹Die Bildung der Vorverarbeitungsregeln werden anhand des Beispiels im Kapitel 4 näher erläutert.

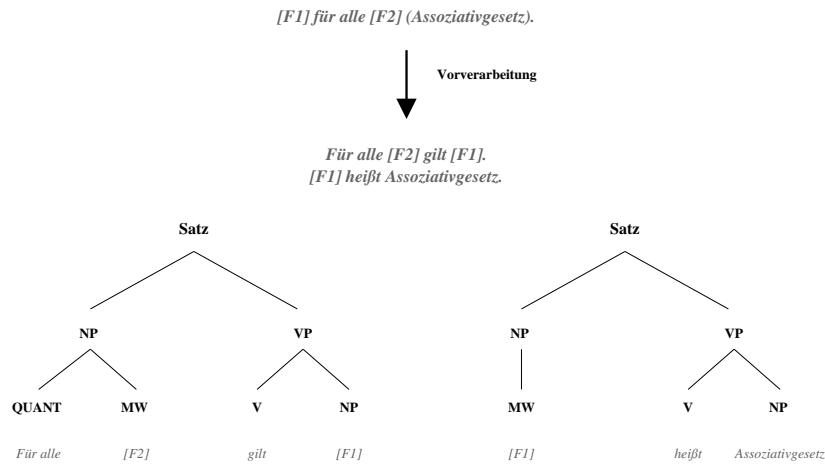


Abbildung 3.19: Syntaktische Analyse des Beispiels 3.3.16 — Teil 2

[Wue95, S.86]

In diesem Fall wird die linke Seite des Satzes als Definiendum angenommen. Der Satz auf der rechten Seite ist das Definiens und muss weiter analysiert werden.

Weitere Beispiele für Definitionen sind:

Beispiel 3.3.18 (Dimension eines Vektorraums)

Ist V ein K -Vektorraum, so definieren wir

$$\dim_K(V) = \begin{cases} \infty, & \text{falls } V \text{ keine endliche Basis besitzt,} \\ r, & \text{falls } V \text{ eine Basis der Länge } r \text{ besitzt.} \end{cases}$$

$\dim_K V$ heißt die Dimension von V über K . [...] [Fis00, S.86]

Beispiel 3.3.19 (Zeilenrang und Spaltenrang einer Matrix)

Für eine Matrix $A \in M(m \times n; K)$ sei

$$\begin{aligned} \text{Zeilenrang } A &= \dim ZR(A) \text{ und} \\ \text{Spaltenrang } A &= \dim SR(A) \end{aligned}$$

[Fis00, S.91]

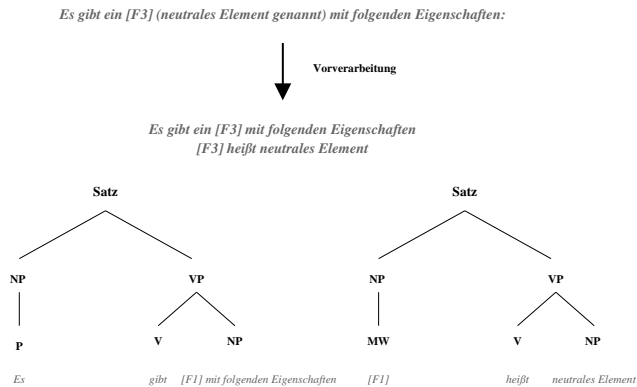


Abbildung 3.20: Syntaktische Analyse des Beispiels 3.3.16 — Teil 3

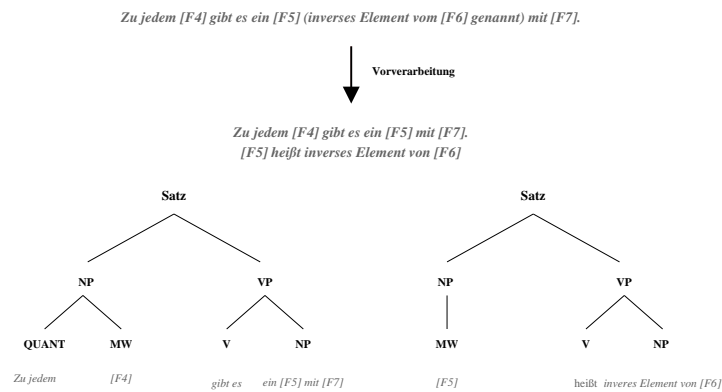


Abbildung 3.21: Syntaktische Analyse des Beispiels 3.3.16 — Teil 4

Beispiel 3.3.20 (Reihe)

Seien $c \in \mathbb{Z}$, \mathbb{Z}_c ein Zahlenabschnitt, und $\{a_z\}_{z \in \mathbb{Z}_c}$ eine Folge. Dann wird der Ausdruck

$$\sum_{k=c}^{\infty} a_k$$

Reihe genannt. [Wue95, S.119]

Beispiel 3.3.21

Für $a > 0, b \in \mathbb{R}$ ist die allgemeine Potenz definiert durch $a^b := e^{b \log a}$. [Wue95, S.186]

Beispiel 3.3.22

Unter einer *Stammfunktion* zu einer Funktion f auf einem Intervall I verstehen wir eine Funktion $F : I \longrightarrow \mathbb{C}$ wie folgt:

- (i) F ist stetig;
- (ii) F ist außerhalb einer höchstens abzählbaren „Ausnahme“-Menge $A \subset I$ differenzierbar, und für alle $x \in I \setminus A$ gilt $F'(x) = f(x)$.

[Kö01, S.166]

Beispiel 3.3.23

Wir nennen eine Funktion $f : I \longrightarrow \mathbb{C}$ fast überall stetig differenzierbar, wenn sie eine Stammfunktion einer Regelfunktion auf I ist. [Kö01, S.202]

Satztypen. In der mathematischen Sprache sind Sätze bewiesene Aussagen, die den Wahrheitswert wahr besitzen. Die gesamte Mathematik besteht aus einer Menge von Aussagen, die sich aus Axiomen logisch ableiten lassen. Im Kapitel 3.3.2 wurde gezeigt, dass Sätze aus einer Voraussetzung und einer Aussage bestehen. Es existieren drei verschiedene Typen von Aussagen: einfache Sätze, Sätze, die Implikationen beschreiben, und Sätze, die Äquivalenzen beschreiben. Die typischen Standardkonstruktionen tauchen daher wieder in den Sätzen auf: *wenn [...] dann* usw. Diese wurden im Kapitel 3.3.3 betrachtet. Allerdings wurden dort nur die singulären Eigenschaften aufgelistet. Die Verwendung von Standardkonstruktionen in Sätzen kann durch die zusätzliche Verwendung von logischen Operatoren und Quantoren komplex werden.

Wie in den vorherigen Abschnitten festgestellt wurde, weisen die verschiedenen Satzentitäten — Fundamentalsätze, Theoreme, Sätze, Propositionen, Korollare, Lemmata und Hilfssätze — keine charakteristischen Unterschiede auf. Daher sollen diese in der weiteren Betrachtung gemeinsam analysiert werden.

Beispiel 3.3.24 (Fundamentalsatz der Algebra)

Jedes Polynom $f \in \mathbb{C}[t]$ mit $\deg f > 0$ hat mindestens eine Nullstelle. [Fis00, S. 63]

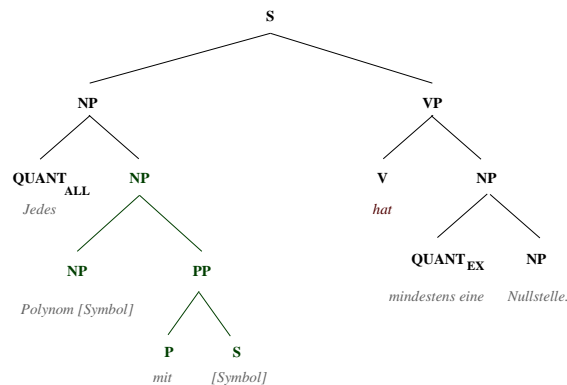


Abbildung 3.22: Syntaktische Analyse des Fundamentalsatzes der Algebra

Analyse des Beispiels 3.3.24. In Abbildung 3.22 wird die syntaktische Analyse des Fundamentalsatzes dargestellt. Die Struktur des Satzes ist dabei relativ einfach. Unter anderem treten in diesem Satz der Allquantor „*jedes*“ und der Existenzquantor „*mindestens eine*“ auf. Etwas komplexer ist der in der Abbildung grün markierte Bereich. Eine semantische Analyse lässt sich dafür nicht mehr so einfach nachvollziehen. Im Kapitel 3.4 soll dies näher betrachtet werden.

Beispiel 3.3.25 (Theorem)

Jeder Vektorraum besitzt eine Basis. [Fis00, S. 83]

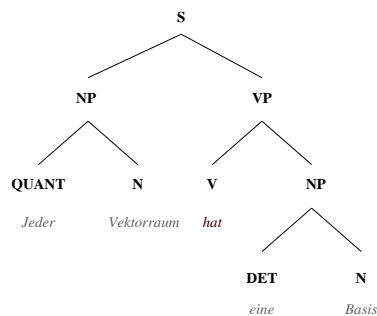


Abbildung 3.23: Syntaktische Analyse des Theorems

Beispiel 3.3.26 (Basisauswahlsatz)

Sei B eine Menge von Vektoren des Vektorraums V . B ist genau dann eine Basis, wenn B ein minimales Erzeugendensystem ist. (nach [Beu94, S. 59])

B ist genau dann eine Basis, wenn B ein minimales Erzeugendensystem ist.

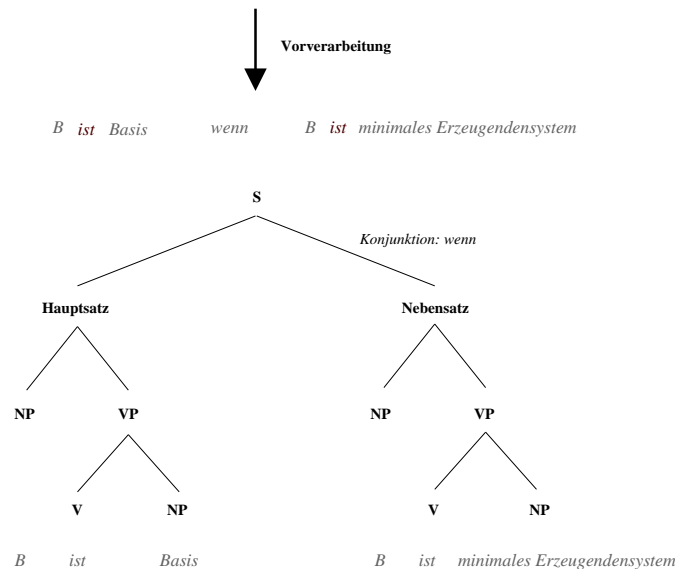


Abbildung 3.24: Analyse des Basisauswahlsatzes

Analyse des Beispiels 3.3.26. Die syntaktische Analyse des Basisauswahlsatzes ist etwas komplizierter (Abbildung 3.24). Durch Transformationen wird der Satz in seine Einzelbestandteile zerlegt, Phrasen ersetzt und Verben umsortiert. Dadurch entstehen strukturierte Satzbausteine, die syntaktisch nach dem Chomsky-Modell (Kapitel 2.4) analysiert werden können.

Beispiel 3.3.27 (Satz ohne Namen)

Der Lösungsraum des linearen Gleichungssystems $A \cdot x = b$ ist genau dann nicht leer, wenn $\text{rang } A = \text{rang } (A, b)$. [Fis00, S. 123]

Beispiel 3.3.28 (Korollar)

Sei K ein beliebiger Körper, $f \in K[t]$ ein Polynom und k die Anzahl der Nullstellen von f . Ist f vom Nullpolynom verschieden, so gilt $k \leq \deg f$. [Fis00, S.62]

Beispiel 3.3.29 (Lemma)

Jede konvergente Folge ist beschränkt. [Kö01, S. 46]

Beispiel 3.3.30 (Proposition)

Für $A \in M(m \times n; K)$ und $b \in K^m$ sind folgende Bedingungen gleichwertig:

1. Das lineare Gleichungssystem $A \cdot x = b$ ist eindeutig lösbar.
2. $\text{rang } A = \text{rang } (A, b) = n$.

[Fis00, S. 127]

3.3.4 Wortebene

Wie auch in anderen Fachsprachen (Kapitel 2.4.4) ist ein wesentliches Merkmal der Mathematik ihr Wortschatz. Die Vielfalt und die Variationsbreite der Wörter ist groß. So wächst der Wortschatz der Mathematik täglich durch neu veröffentlichte Publikationen. Durch die strenge Strukturierung und den Anspruch, eine exakte Wissenschaft zu sein, lässt sich diese Vielfalt strukturell organisieren (Kapitel 3.4) und somit eine Terminologie der Mathematik aufbauen.

Bei der Betrachtung des Wortschatzes kann zwischen außerfachsprachlichen Wörtern und Wörtern, die nur in der betrachteten Fachsprache verwendet werden, unterschieden werden [GHL02, S. 618]. So verwendet auch die Mathematik Begriffe aus dem alltäglichen Wortschatz, wie z.B. *Knoten*, *Quelle*, *Wurzel*, *Raum* usw., die jedoch fachspezifisch angewendet werden. Dies beinhaltet zwar Gefahren für den ungeübten mathematischen Laien, der den Begriffen eine allzu anschauliche Bedeutung beimisst. Jedoch besteht bei einer computergestützten Analyse keine solche Gefahr, wenn diese nur für die Fachsprache durchgeführt und somit auf die Domäne reduziert wird. Eine computergestützte linguistische Analyse hat keine Anschauung über die Bezeichnungen von mathematischen Objekten. Probleme treten erst dann auf, wenn einem Begriff mehrere Bedeutungen zugeordnet werden können.

Problematisch ist in der Mathematik, dass einem Sachverhalt oder dem Gegenstand der Betrachtungen mehreren Begriffen zugeordnet werden können. So beschreiben die Bezeichnungen *Multiplikation mit Skalaren* und *skalare Multiplikation* denselben Sachverhalt. Vielfach werden alternative Bezeichnungen genannt, indem sie hinter dem eigentlichen Begriff in Klammern gesetzt werden,

wie z.B. „*Multiplikation mit Skalaren (auch skalare Multiplikation genannt)*“. Schwieriger wird es jedoch, wenn sich in einer Entität die Bezeichnung für einen Sachverhalt ändert:

Beispiel 3.3.31 (...)

b) Das *inverse Element* a' ist eindeutig und hat auch die Eigenschaft $a \cdot a' = e$ für alle $a \in G$.

c) Da das *Inverse* nach b) eindeutig bestimmt ist, kann man es mit a^{-1} bezeichnen. [...].

[Fis00, S. 45]

Die Begriffsbildung in der Mathematik ist sehr einfach. Eine terminologische Bezeichnung besitzt meistens eine eindeutige Zuordnung. So lassen sich einige der oben genannten Konstruktionen auflösen, indem standardisierte Konstruktionsmöglichkeiten vorgegeben werden, wie z.B. *Kantengraph* und *Graph* [Artikel/Präposition] *Kante*. Beide Konstruktionen können dann synonym verwendet werden. Die Verwendung von zwei Substantiven zur Wortbildung (Komposition) tritt häufig in anwendungsbezogener mathematischer Literatur auf.

In der Mathematik treten häufig Fremdwörter aus dem Griechisch-Lateinischen auf, wie z.B. „*linear*“ (lat.: *lineāris*), *äquivalent* (lat.: *aequivalēns*), *Geometrie* (gr.-lat.: *geōmetría*) usw.

Häufig werden Eigennamen für die Entdecker von Sachverhalten verwendet. So gibt es z.B. das *Lemma von Zorn*, die *Cauchy-Folge* und den *Hilbertraum*. Selten tritt dagegen die Verwendung von Symbolen und Ziffern in Wörtern auf, z.B. in *L-quadratintegrierbar*, *1.Ordnung* und *ε-Umgebung*. Akronyme treten z.B. in der Lineare Algebra und Analysis fast nicht auf.

In der mathematischen Sprache werden häufig Mehrwortbezeichnungen verwendet. Dabei wird meistens ein Adjektiv mit einem Substantiv verbunden, so dass eine feststehende Nominalphrase konstruiert wird, die einen Sachverhalt beschreibt, z.B. *stetige Funktion*, *leere Menge*, *geometrische Reihe* usw. Diese Adjektive schränken damit den Geltungsbereich des Substantivs ein. In diesem Zusammenhang ist das Negationspartikel *nicht* zu nennen, das mit dem Adjektiv in Verbindung tritt, z.B. *nicht leere Menge*. Allerdings gibt es für diese Wortverbindungen verschiedene Schreibweisen, wie *nichtleere Menge* oder

nicht-leere Menge. Weitere Mehrwortbezeichnungen entstehen durch präpositionale Ergänzungen wie z. B. *Äquivalenzrelation auf der Menge X* , *Homomorphismus von Gruppen*, wodurch eine enge semantische Beziehung zwischen Verb und Objekt des Satzes ausgedrückt wird, sowie durch die Verwendung des Genitives possessivus (*Teilmenge K eines metrischen Raumes M*). Bei der Bezeichnung eines Sachverhaltes durch Adjektive können zwei Adjektive auch hintereinander angeordnet werden, z. B. *absolut konvergent*, *linear unabhängig*.

Die Komparation von Adjektiven erfolgt nicht. So gibt es nicht die Steigerung – surjektiv – surjektiver – am surjektivsten. Adverbien werden nur in eingeschränktem Maße verwendet. So werden modale Adverbien (*gern*, *unverzüglich* usw.) nicht verwendet. Dagegen werden kausale Adverbien eingesetzt, z. B. *andernfalls*, *somit* usw. Die Ursache dafür ist, dass zahlreiche kausale Adverbien auch konjunktionale Funktionen übernehmen können, die durch Implikationsaussagen verursacht werden.

So ist der Begriffs- und Wortbildungsprozess sehr viel eingeschränkter als in anderen Fachsprachen, die deutlich mehr Möglichkeiten bieten, um Begriffe und Wörter zu konstruieren [GHL02, S. 619ff]

3.3.5 Symbolebene

Die mathematische Sprache enthält eine reichhaltige Sammlung von mathematischen Zeichen. Diese Zeichen und Symbole können einerseits direkt hinter einem Begriff stehen, wie z. B. *Ring R* , *Menge M* , andererseits werden sie als äquivalente Bezeichnung für den entsprechenden Begriff verwendet. So wird das Symbol \mathbb{R} für den Begriff *reelle Zahlen* eingesetzt.

Nicht nur einzelne Zeichen werden so verwendet, sondern auch Terme und Formeln. So tritt im Text häufig die Bezeichnung $x \in X$ auf für „ *x ist ein Element von X* “. Wenn V ein Vektorraum ist, dann ist die Bezeichnung $v \in V$ gleichbedeutend mit „ *v ist ein Element des Vektorraumes*“ oder „ *v ist ein Vektor*“.

Diese Terme und Formeln können jedoch sehr kompliziert aufgebaut sein. So ist der folgende Satz für eine rein linguistische Analyse schwer zugänglich: „[...] Eine Menge $\mathbb{Z}_c := \{n \mid n \in \mathbb{Z}, n \geq c\}$ heißt *Zahlenstrahl* [...]“. In diesem Fall müsste zumindest erkannt werden, dass \mathbb{Z}_c eine synonyme Bezeichnung für den

Einige wichtige Symbole und ihre Auswertung

$a \in A$	a ist ein Element von A
$\forall a$	Für alle a
$\exists a$	Es existiert ein a
$A := B$	Definiens B und Definiendum A
$A \Leftrightarrow B$	A genau dann, wenn B
\mathbb{R}, \mathbb{N} usw.	reelle Zahlen, natürliche Zahlen
$f : A \longrightarrow B$	f ist eine Abbildung von A nach B

Zahlenstrahl ist. Auch die Bezeichnung „Abbildung $f : A \longrightarrow B$ “ ist problematisch, da es unter Umständen wichtig ist, welche Mengen A und B bezeichnen. Daher ist es notwendig, zumindest die fundamentalsten mathematischen Zeichen zu erkennen und zu analysieren, um sie damit einer linguistischen Analyse zugänglich zu machen.

3.3.6 Zusammenfassung

In diesem Abschnitt sollen die in den vorherigen Abschnitten beschriebenen Erkenntnisse zusammengefasst werden. Es wird aus den aufgeführten Gründen folgendes Modell für die Analyse der mathematischen Sprache folgendes Modell vorgeschlagen:

- 1. Schritt:** Erkennung der Entitätenstruktur.
- 2. Schritt:** Auflösung der Binnenstruktur.
- 3. Schritt:** Syntaktische Auflösung der Sätze.
- 4. Schritt:** Analyse einzelner Satzbausteine

Erkennung der Entitäten. Die Entitäten müssen zuerst als solche erkannt werden, damit weitere Analyseschritte getrennt für jeden Entitätentypen erfolgen können. Bei dieser Erkennung treten aber Schwierigkeiten auf. So wurde im Rahmen dieser Arbeit festgestellt, dass es keine charakteristischen Merkmale

für die einzelnen Satzentitäten gibt. Daher können diese nur durch die explizite Angabe des Entitätentyps korrekt erkannt werden. Eine weitere mögliche Entitätentyperkennung kann durch die Bestimmung der Position im Textgefüge erfolgen. Dies ist jedoch mit einigen Unsicherheiten behaftet. Glücklicherweise kennzeichnen jedoch Mathematiker häufig die Sätze mit der expliziten Angabe des Entitätennamens.

Auflösung der Binnenstruktur. Die Auflösung der Binnenstruktur hat viele Vorteile. Der wichtigste Schritt ist dabei die Trennung der Voraussetzungen von der eigentlichen Aussage der untersuchten Entität. In den Voraussetzungen werden u. a. zahlreiche Bezeichnungen und Begriffe festgelegt, die für eine semantische Analyse der darauf folgenden Aussage von Bedeutung sind. Die Voraussetzungen sind dabei meist gut zu erkennen, so dass die Binnenstrukturen leicht zuzuordnen sind.

Am einfachsten ist diese Auflösung für Definitionen, da in deren Aussageanteil praktisch nur eine einzige Form der Binnenstruktur vorkommt. Für Sätze hat sich gezeigt, dass es lediglich drei Satzstrukturen gibt, nämlich den einfachen Aussagesatz, die Implikation und die Äquivalenz. Diese drei Typen lassen sich aufgrund ihres klaren Aufbaus relativ leicht erkennen. Axiome dagegen sind problematisch, da sie keine einheitliche Binnenstruktur aufweisen.

Syntaktische Analyse der Sätze. Die syntaktische Analyse von Sätzen kann nach dem in Kapitel 2.4 vorgestellte Modell von Chomsky erfolgen. Trotz seiner Einfachheit ist es dafür ausreichend, da mathematische Sätze in Lehrbüchern weniger komplexe Strukturen aufweisen als Sätze in der Alltagssprache. Des Weiteren wurde festgestellt, dass die syntaktische Struktur sehr stark durch die Prädikatenlogik beeinflusst wird und somit meist einfache Satzstrukturen auftreten.

Einige Satzstrukturen sind jedoch auf den ersten Blick nicht so einfach zu analysieren. Sie können allerdings durch Vorverarbeitungsregeln in eine analysierbare Form überführt werden. Diese Vorverarbeitungsregeln wurden aus der Transformationsgrammatik nach Chomsky entwickelt. Sie ergeben sich aus einer relativ geringen Anzahl feststehender Phrasen, die sich in Form eines syntaktischen

Lexikons für eine Analyse zusammenfassen lassen. Ein Beispiel dafür ist die Binnenstruktur *Voraussetzung*.

Analyse der Satzbausteine. Die bei der Analyse nach Chomsky entstehenden syntaktischen Strukturen weisen für mathematische Texte einige Besonderheiten auf. So verwendet die mathematische Sprache Symbole und Formeln, die direkt hinter dem Substantiv auftreten. Daher ist es sinnvoll, Symbole als einen Teil der Nominalphrase zu sehen:

$$\text{NP} ::= [\text{DET}] [\text{ADJ}] \text{N} [\text{SYM}]$$

Des Weiteren werden Quantoren teilweise wie Artikel verwendet und können in Nominalphrasen an deren Stelle treten.

$$\text{NP} ::= [\text{DET/QUANT}] [\text{ADJ}] \text{N} [\text{SYM}]$$

Es treten häufig rekursive Nominalphrasenstrukturen der Form

$$\text{NP} ::= \text{NP NP}$$

auf, die beliebig mit weiteren Nominal- und Präpositionalphrasen verschachtelt sein können. Deren syntaktische Analyse ist einfach, die semantische Analyse jedoch außerordentlich komplex. Präpositionen hinter einer Nominalphrase schränken diese in ihrem Wirkungsbereich ein. Diese Einschränkung wird auch von Adjektiven für ein darauf folgendes Substantiv erreicht.

In der mathematischen Sprache werden definierende Bezeichnungen in Eigenschaften häufig in Klammern hinter die betreffende Symbolfolge bzw. Aussage gesetzt. Dies muss bei einer syntaktischen wie semantischen Analyse berücksichtigt werden. Des Weiteren treten hauptsächlich bei Aufzählungen Sätze ohne Verbformen auf, wie z. B. *f injektiv*. Dies muss ebenfalls berücksichtigt werden.

Eine semantische Analyse wurde bis jetzt nicht beschrieben. Das Problem besteht darin, dass die semantischen Informationen einer einzelnen Entität ohne Kontextwissen nur einen geringen Informationsgehalt besitzen. Ohne eine Möglichkeit, mathematisches Hintergrundwissen einzubringen, ist eine semantische Analyse daher sinnlos. Dieses Hintergrundwissen soll in dieser Arbeit durch eine Wissensbasis repräsentiert werden (Kapitel 3.4).

3.4 Wissensstrukturen

*„In hierarchischen Strukturen kommt das Gute
nie von oben. Obenauf schwimmt der Abschaum.*

Das Wertvolle ist der Bodensatz. “

(H. A. Pestalozzi, Schweizer Autor,
Manager und Gesellschaftskritiker, 1929-2004)

3.4.1 Ontologie der Mathematik

Einleitung. Im Kapitel 3.3 trat das Problem auf, dass eine semantische Analyse der mathematischen Sprache nicht einfach zu realisieren ist. Mathematische Begriffe und Sachverhalte bilden ein großflächiges Beziehungsnetzwerk. Viele dieser Begriffe und Sachverhalte werden von den Autoren der Texte vorausgesetzt und nicht explizit definiert. Dies ist auch in anderen natürlichsprachlichen Texten der Fall. Allerdings sind die terminologischen Strukturen in der Mathematik klarer und strenger formalisiert, so dass die Begriffe und Sachverhalte genau definiert werden können.

Wissensbasierte Systeme unterstützen die Organisation und Speicherung von Begriffen und Sachverhalten. Gleichzeitig sind diese gespeicherten Informationen jederzeit wieder abrufbar. Als Repräsentationsmechanismus für diese Speicherung eignen sich semantische Netze. Diese werden häufig bei der Untersuchung der Semantik von natürlichsprachlichen Texten verwendet, wie z. B. in MultiNet [Hel00]. Durch den strukturierten Aufbau der mathematischen Terminologie sollte auch bei der semantischen Analyse von mathematischen Texten eine solche Konzeption hilfreich sein.

Zur Beschreibung von semantischen Netzen in wissensbasierten Systemen werden heutzutage als formaler Beschreibungsmechanismus Ontologien verwendet. Ontologien sind dabei reichhaltiger strukturiert als semantische Netze. Sie verwenden Schemata, die im Kapitel 2.2.2.2 als Erweiterung von semantischen Netzen eingeführt wurden.

Der Vorteil der Verwendung von Ontologien liegt in der Erzeugung eines Systems, das technisch umsetzbar und somit als Modell überprüfbar ist. Daher

soll auch für die Mathematik eine Ontologie entworfen werden, die insbesondere Wissenstrukturen formalisiert, die durch eine vorherige semantische Analyse aus mathematischen Texten extrahiert werden.

In den Kapiteln 2.2.1 und 2.2.3 wurde kurz auf die Struktur von Ontologien eingegangen. Die Grundbausteine einer Ontologie sind Konzepte und Beziehungen²⁰ zwischen diesen Konzepten. Daher muss für die Mathematik eine geeignete Konzeptualisierung gefunden werden. Unter einer Konzeptualisierung wird ein abstraktes Modell verstanden, das mathematische Strukturen abbilden kann.

In der hier zu betrachtenden mathematischen Ontologie sollen grundlegende mathematische Strukturierungsprinzipien verwendet werden. Als Fundament dienen die folgenden Beschreibungsmechanismen (Kapitel 3.2.2 und 3.2.3):

- **Prädikatenlogik und axiomatische Mengenlehre**
- **Grundstrukturen nach Bourbaki**

Prädikatenlogik und axiomatische Mengenlehre. Durch die Verwendung der axiomatischen Mengenlehre (Kapitel 3.2.3) kann ein abstraktes mathematisches Ontologiemodell konstruiert werden. Die axiomatische Mengenlehre beruht auf dem allgemeinsten Begriff in der Mathematik: *der Menge*. Der Aufbau der Mengenlehre entspricht der einer üblichen mathematischen Theorie, allerdings ist ihre Begriffs- und Sachverhaltsbildung wegweisend für „*praktisch alle*“ [Ebb94, S. vii] mathematischen Fachdisziplinen. So bildet die Mengenlehre u. a. ein terminologisches Modell, aus dem sich viele mathematische Begriffe ableiten lassen. Die Prädikatenlogik dient dabei als Beschreibungssprache der Mengenlehre.

Im folgenden Abschnitt soll ein einfaches ontologisches Modell auf Grundlage der axiomatischen Mengenlehre konstruiert werden. Dieses Modell erhebt keinen Anspruch auf Vollständigkeit. In den weiteren Schritten soll dieses Modell für die semantische Analyse von einfachen mathematischen Entitäten verwendet werden, um die extrahierten Wissenstrukturen kontextuell zu binden. Es wird

²⁰Beziehungen werden auch als *Relationen* bezeichnet. Allerdings wird der Begriff Relation schon in der Mathematik verwendet. Daher wird in dieser Arbeit der Begriff Beziehung bevorzugt, um Verwechslungen zu vermeiden.

sich herausstellen, dass die konstruierte Ontologie einige Fragen aufwirft, die in Kapitel 5 genauer diskutiert werden.

Konzepte in einer Ontologie fassen Objekte zusammen, die gemeinsame Eigenschaften²¹ aufweisen. Die axiomatische Mengenlehre nach Zermelo-Fraenkel beschreibt das fassbare, grundlegende ontologische Konzept MENGE. Dabei definiert sie nicht den Begriff der Menge, sondern beschreibt die Beziehungen zwischen Mengen, ohne irgendeine Angabe zu machen, was dieses Konzept beinhaltet. Aufgrund des Extensionalitätsaxioms (Axiom 3.2.3) kann davon ausgegangen werden, dass alle mathematischen Objekte sich auf den Mengenbegriff zurückführen lassen [Ebb94].

Das gilt jedoch z. B. nicht für das Elementzeichen \in und die Teilmengenbeziehung \subset . Diese sind keine Mengen im axiomatischen Sinne. In der Ontologie entsprechen sie daher den Beziehungen zwischen den Konzepten: *is_element_of* und *is_subset_of*.

Durch das Aussonderungsaxiom (Axiom 3.2.9) wird ein Konstruktionsprinzip gegeben, durch das „Untermengen“ gebildet werden können, die sich durch gewisse Eigenschaften auszeichnen. Dies entspricht der Bildung neuer Konzepte aus dem Konzept MENGE durch eine *is_a* Beziehung²², die dann noch zusätzliche Eigenschaften besitzen.

Es gibt bestimmte Objekte in der Mathematik, die einen grundlegenden Charakter besitzen, wie z. B. Relationen und Funktionen. Für die Konzeption dieser Objekte auf der Grundlage des Konzepts MENGE muss der Begriff des *geordneten Paares* (Definition 3.2.7) eingeführt werden. In der axiomatischen Mengenlehre wird dies als Paarmengenbildung (Axiom 3.2.4) mit der speziellen Eigenschaft angesehen, dass die Reihenfolge der Objekte eindeutig festgelegt ist. Somit ist auch das geordnete Paar eine spezielle Menge. Der Zuordnungsmechanismus, der dem geordneten Paar den linken $(*, y)$ bzw. den rechten Eingang $(x, *)$ zuordnet, wird als Projektionsoperation [Ebb94, S. 55] bezeichnet. Dies soll nicht direkt in dieser Ontologie berücksichtigt werden (Kapitel 5).

²¹Eigenschaften werden auch als Attribute bezeichnet.

²²Sie wird auch SUB-Relation genannt. „Alle einem gemeinsamen Oberbegriff vermöge der SUB-Relation untergeordneten Begriffe bilden zusammen mit ersterem eine Begriffshierarchie und konstituieren damit einen Teilbaum im SN [semantischen Netz]. Der an der Spitze der Hierarchie stehende Knoten ist eben dieser Oberbegriff. Die terminalen individuellen Begriffsknoten des Baums heißen Instanzen.“ [Hel00]

Aus dem geordneten Paar lässt sich induktiv das Konzept TUPEL ableiten, das den mathematischen n -Tupeln entspricht. Es ist über eine *is_a*-Beziehung mit dem Konzept GEORDNETESPAAR verbunden, das wiederum über eine *is_a*-Beziehung direkt mit dem Konzept MENGE verknüpft ist (Abbildung 3.25).

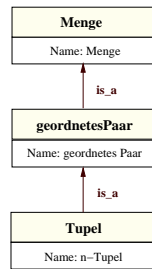


Abbildung 3.25: Darstellung eines Tupels

Das Konzept GEORDNETESPAAR wird vom *kartesischen Produkt* (Definition 3.2.8) verwendet. Wie das Tupel bildet auch das Konzept KARTESISCHESPRODUKT eine spezielle Menge (*is_a*-Beziehung). Jedoch wird zusätzlich die Eigenschaft gefordert, dass das kartesische Produkt das Konzept GEORDNETESPAAR verwendet. Dies wird durch eine neue Beziehung *has_part* realisiert, die besagt, dass ein Konzept (in diesem Fall GEORDNETESPAAR) Bestandteil eines anderen Konzeptes (in diesem Fall KARTESISCHESPRODUKT) ist (Abbildung 3.26).

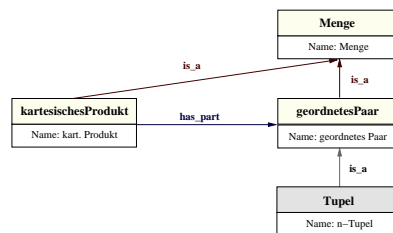


Abbildung 3.26: Darstellung eines kartesischen Produktes

Somit kann nun das Konzept RELATION (Definition 3.2.10) in das bestehende Modell eingefügt werden. Es besteht eine *has_part*-Beziehung vom Konzept RELATION zum kartesischen Produkt (Abbildung 3.27).

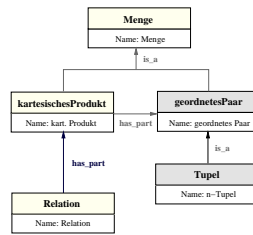


Abbildung 3.27: Darstellung einer Relation

Die Abbildung (Definition 3.2.13) leitet sich durch eine *is_a*-Beziehung vom Konzept RELATION ab (Abbildung 3.28).

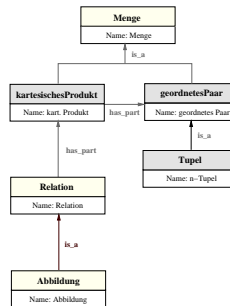


Abbildung 3.28: Darstellung der Abbildung

Abbildung 3.29 zeigt eine in Teilen vereinfachte Form der bisher beschriebenen Ontologie, die um die Konzepte der ASSOZIATIVITÄT, KOMMUTATIVITÄT und DISTRIBUTIVITÄT erweitert wurde. Diese stammen aus der Aussagenlogik und spielen bei der Formulierung von Eigenschaften anderer Konzepte eine wichtige Rolle. Sicherlich ist das nicht die einzig mögliche Struktur einer Ontologie der Mathematik, aber sie ist für die im *mArachna-Projekt* behandelten Fragestellungen sinnvoll und ausreichend.

Beschreibung durch Bourbaki. Neben Hilbert gilt als wesentlicher Begründer und Befürworter der axiomatischen Theorie eine Gruppe französischer Mathematiker, die ihre Forschungsergebnisse ab 1935 unter dem Pseudonym Nicolas Bourbaki veröffentlichte. Bourbaki erweiterte den proklamierten Ansatz

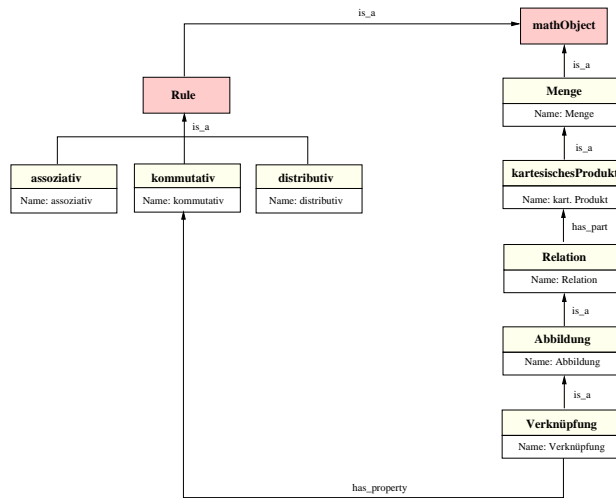


Abbildung 3.29: Konzeption mathematische Ontologie

von Hilbert. Sie verstanden die Mathematik als ein geordnetes Zusammenspiel von Strukturen wie Mengen, Gruppen, Körpern usw. Dabei existieren Kerngebiete, die für alle mathematischen Disziplinen unerlässlich sind. Dies sind die Mengenlehre, die Algebra, die Topologie und die Analysis. Als Fundament dient die Logik und die Mengenlehre [Bou74].

Bourbaki definiert drei grundlegende Strukturierungselemente:

- **Ordnungsstrukturen**
- **algebraische Strukturen**
- **topologische Strukturen**

So lassen sich mathematische Objekte über Mengen mit gewissen Strukturen beschreiben. Eine Addition $+$: $M \times M \longrightarrow M$ kann mittels des geordneten Paares $(M, +)$ beschrieben werden. Die Addition ist dabei das strukturbildende Element auf der Menge M . Der Begriff der Gruppe wird durch das 2-Tupel $(G, *)$ beschrieben, wobei die Gruppe als Menge mit der strukturbildenden Verknüpfung $*$ aufgefasst wird.

Das Konzept von Bourbaki lässt sich jedoch nicht ohne Aufwand auf die Analyse der mathematischen Sprache übertragen. Der Hauptgrund dafür ist,

dass Autoren in Lehrbüchern dieses Konzept nicht durchgängig verwenden. Dadurch ist in vielen Definitionen und Sätzen der obige typische Konstruktionsmechanismus nicht vorhanden, so dass die in den Texten enthaltenen semantischen Strukturen zunächst in eine Struktur nach Bourbaki übersetzt werden müssten. Aus diesem Grund werden strikte Bourbaki-Strukturen in dieser Arbeit nicht verwendet. Sie könnten sich in Zukunft jedoch als nützlich für die Darstellung größerer mathematischer Konzepte wie z. B. ganzer Disziplinen erweisen.

Aus diesen Betrachtungen wird deutlich, dass sich die verwendete einfache Ontologie nur ungenügend für die Beschreibung einiger mathematischer Konzepte eignet. Für die im Rahmen dieser Arbeit durchgeführten Analysen ist sie jedoch ausreichend. Weiterführende Ansätze werden im Kapitel 5 besprochen. Dabei wird unter anderem das Konzept eines semantischen Hypergraphen diskutiert.

3.4.2 Taxonomie der Mathematik

Im ersten Abschnitt dieses Kapitels wurde eine Ontologie auf Basis der axiomatischen Mengenlehre entwickelt, die die Grundlage für die weiteren Betrachtungen bilden soll. Die Taxonomie der Mathematik beschreibt die Hierarchie der Begriffe und Sachverhalte der Mathematik und modelliert so ihre Strukturen. In den folgenden Abschnitten soll anhand von einfachen Beispielen erläutert werden, wie mathematische Wissensstrukturen aus Entitäten extrahiert und in die konstruierte Ontologie eingeordnet werden können.

Vorbetrachtungen. Die Mathematik besteht aus einer Reihe von mathematischen Fachdisziplinen, wie z. B. der Geometrie, der Arithmetik und der Zahlentheorie. Dabei können über 3000 spezialisierte Einzeldisziplinen der Mathematik [DHM95, S. 20] unterschieden werden.

„The Mathematics Books at Brown University are housed on the fifth floor of the Science Library. In the trade, this is commonly regarded as a fine mathematical collection, and a rough calculation shows that this floor contains the equivalent of 60,000 average-sized volumes. [...] This amount of knowledge and information is far beyond the comprehension of any one person.“ [DHM95, S. 17]

es leicht ersichtlich, dass eine solche Übersicht durch Hinzunahme weiterer Fachdisziplinen schnell schwer überschaubar wird. Völlig unmöglich wird der Gedanke aber erst, wenn in den Fachdisziplinen auch noch Unterstrukturen wie z. B. grundlegende Konzepte einer Disziplin dargestellt werden sollen. Angesichts von über 3000 Einzeldisziplinen erscheint dies utopisch.

Die ersten Schritte werden daher auf Fachdisziplinen — die lineare Algebra und Algebra — beschränkt. Die Algebra nimmt eine zentrale Stellung in der Mathematik ein und bildet somit die Grundlage für viele mathematische Fach-

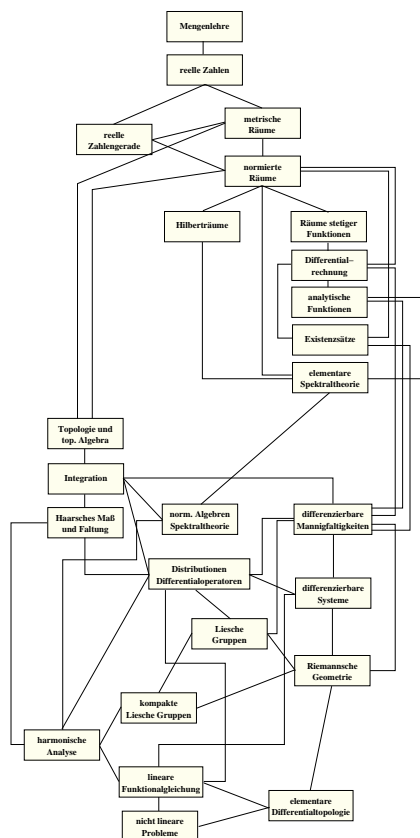


Abbildung 3.30: Ausschnitt der mathematischen Fachgebiete [Die75]

disziplinen. Die Algebra lässt sich nach Bourbaki folgendermaßen strukturieren (sieht auch Abbildung 3.31):

1. **Algebraisch strukturierte Mengen:** Gruppe, Ring, Körper, Vektorraum usw.
2. **Algebraische Strukturen werden induziert durch:** innere und äußere Verknüpfung mit speziellen Eigenschaften: kommutativ, assoziativ, distributiv usw.
3. **Algebraisch strukturierte Untermengen:** Untergruppe, Unterring, Ideal, Quotientenstruktur usw.
4. **Strukturerhaltende Abbildungen:** Homomorphismus, Isomorphismus, lineare Abbildung usw.

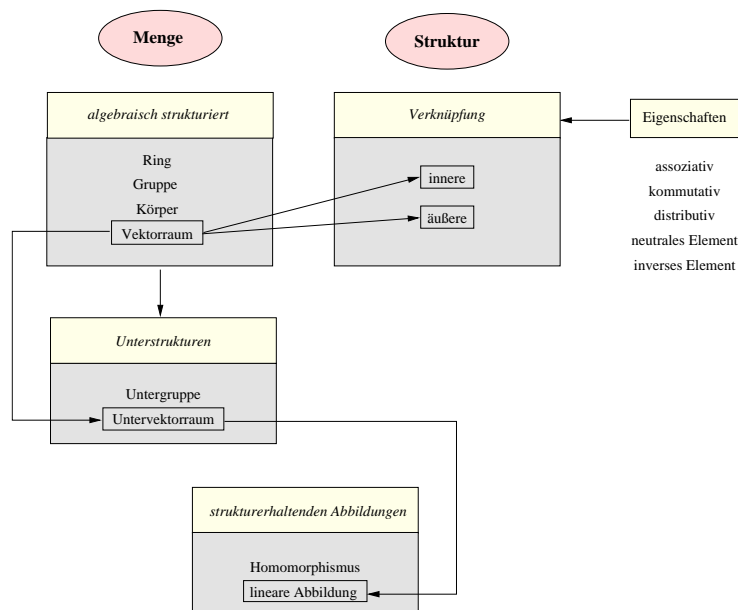


Abbildung 3.31: Überblick Algebra

In Abbildung 3.31 ist eine Unterstrukturierung abgebildet. Diese deutet auf die lineare Algebra hin, bei der der Begriff Vektorraum eine zentrale Rolle spielt.

Taxonomie der Entitäten. Informationen werden in der Mathematik durch die in Kapitel 3.3.1 diskutierten fundamentalen Entitäten *Axiom*, *Definition* und *Satz*²³ dargestellt. Daher soll als erstes anhand eines Beispiels gezeigt werden, wie aus Definitionen semantische Informationen extrahiert werden. Anschließend sollen Sätze betrachtet werden.

Die Entität *Definition*. Eine Definition ist die Zusammenfassung von mathematischen Begriffen und Sachverhalten, um einen neuen Begriff zu beschreiben. Im Allgemeinen treten in der Mathematik nur abstrakte Begriffe ohne direkte Entsprechung in der Alltagssprache auf. Daher werden in einer axiomatisch aufgebauten Mathematik *alle* Begriffe definiert. Diese Begriffe bilden dann ein grundlegendes Begriffsgerüst. Die Grundidee der semantischen Analyse ist es, Begriffe, die durch Definitionen beschrieben werden, nur dann in die Wissensbasis zu integrieren, wenn die Sachverhalte und Begriffe, durch die sie definiert werden, schon in der Wissensbasis vorhanden sind. Eine andere Herangehensweise würde die Integrität der Wissensbasis gefährden.

Die wichtigste Beziehung zwischen den Begriffen ist die *is_a*-Beziehung. Die Eigenschaften des Oberbegriffs werden bei dieser Konstruktion auf den Unterbegriff übertragen. Dabei können die Eigenschaften entweder fest zugeordnete Ausprägungen oder Defaultwerte besitzen²⁴. Das Problem bei der Verwendung von Eigenschaften mit Defaultwerten ist, dass eine solche Strukturierung nicht unbedingt aus der Analyse von Texten gewonnen werden kann. Der Sinn der Wissensbasis ist es jedoch, Wissen aus der Extraktion von semantischen Informationen aus Texten zu integrieren. Daher kann in einem ersten Schritt davon ausgegangen werden, dass Eigenschaften ohne Defaultwerte verwendet werden.

Die Begriffe entsprechen den Konzepten in der Ontologie. Der Konzeptname wird durch den Begriffsnamen festgelegt. In erster Linie wird der neue Begriff durch eine *is_a*-Beziehung direkt mit den im Definiens angegebenen Begriffen

²³Beweise und Beispiele sind nicht Untersuchungsgegenstand dieser Arbeit.

²⁴In Ontologien werden häufig Mechanismen für die Bildung von Prototypen bzw. prototypischen Wissens, d. h. von typischem Wissen mit Defaultangaben verwendet (Kapitel 2.2). Diese sollen zunächst nicht betrachtet werden. Jedoch stellen Prototypen eine interessante Alternative dar, mathematisches Wissen stärker zu strukturieren. So könnte z. B. der Begriff *lineare Algebra* mit den Slots *Definitionsbereich* bzw. *Wertebereich* versehen werden. Die Slots erhalten dann entsprechende Defaultwerte, die belegt werden dürfen.

neu eingebunden. Wenn das Definiendum Eigenschaften aufweist, werden diese Eigenschaften über eine *has_property*-Beziehung an das Definiendum gebunden.

Es soll anhand des Beispiels 3.3.16 eine semantische Analyse exemplarisch durchgeführt werden. Zuerst wird die Definition dafür zerlegt:

Erster Teil

Eine Menge G zusammen mit einer Verknüpfung $$ heißt Gruppe, wenn folgende Eigenschaften erfüllt sind:*

Nach der syntaktischen Analyse wird das Verb *heißen* als Schlüsselwort für eine Definition erkannt. Darauffolgend werden die syntaktischen Komponenten wie z. B. die Nominalphrase semantischen Satzbausteinen — Prädikat, Subjekt oder Objekt — zugeordnet. Dabei entspricht dem Definiendum das Objekt und dem Definiens das Subjekt (siehe Abbildung 3.32). Somit entstehen folgende Zuordnungen:

1. Prädikat ::= Subjekt Objekt
2. Subjekt ::= NP
3. $V ::= \text{Prädikat} \wedge NP ::= \text{Objekt}$ (aus VP)

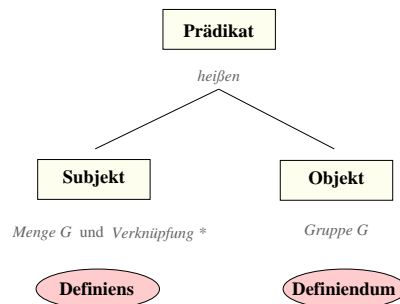


Abbildung 3.32: Zerlegung der Definition in semantische Satzbausteine

Daraus ergibt sich für das Beispiel 3.3.16:

$$[\text{ist, Menge, Gruppe}] \wedge [\text{ist, Verknüpfung, Gruppe}]$$

Das Prädikat *ist* entspricht dann der *is_a*-Beziehung. Der Nebensatz leitet die Eigenschaften ein, d. h. alles, was nach dem Doppelpunkt folgt, wird im folgenden zweiten Teil behandelt und mittels einer *has_property*-Beziehung an das Definiendum gebunden.

Der Nebensatz leitet zu den Eigenschaften hin, d. h. alles was nach dem Doppelpunkt folgt, wird nun durch den zweiten Teil behandelt und mit der *has_property*-Beziehung verknüpft. Allerdings gibt es dabei einige Ausnahmen.

Zweiter Teil

G1 $(a * b) * c = a * (b * c)$ für alle $a, b, c \in G$ (*Assoziativgesetz*)

G2 *Es gibt ein $e \in G$ (neutrales Element genannt) mit den folgenden Eigenschaften:*

- $e * a = a$ für alle $a \in G$
- Zu jedem $a \in G$ gibt es ein $a' \in G$ (*inverses Element von a genannt*) mit $a' * a = e$.

Die Analyse von G1 bereitet einige Schwierigkeiten. Die semantische Information steht ausschließlich in den Formeln, gefolgt von der Bezeichnung *Assoziativgesetz* in Klammern. In der syntaktischen Analyse wird daher der Satz in zwei Sätze zerlegt: Ein Satz beschreibt die mathematische Aussage, der andere den zu definierenden Begriff (Kapitel 3.3). In dieser Form würde dann der Begriff *Assoziativgesetz* direkt an den Begriff *Gruppe* gebunden werden. Bei dem *Assoziativgesetz* handelt es sich aber um eine Eigenschaft, die die Verknüpfung besitzen kann. Dies könnte dadurch abgefangen werden, dass der Begriff *Assoziativgesetz* nur mit dem Konzept *Verknüpfung* eine Verbindung eingehen kann, mit der zusätzlichen Einschränkung, dass sich dies auf das Konzept *Gruppe* bezieht:

- Folgende Eigenschaft ist daher unvorteilhaft:
[Eigenschaft, Assoziativgesetz, Gruppe]
- Folgende Eigenschaft gilt:
[Eigenschaft, Assoziativgesetz, Verknüpfung_{Gruppe}]

Allerdings kann nach dieser Vorgehensweise die Formel $(a * b) * c = a * (b * c)$ nicht analysiert werden. Wie oben beschrieben, ist dies von Nachteil, da eine Analyse der Formel eventuell zu einer Auflösung des Widerspruchs geführt hätte. Das mathematische Zeichen $a, b, c \in G$ ist allerdings auflösbar, da dies in eine sprachliche Form gebracht werden kann: *a, b und c sind Elemente der Gruppe*.

In G2 werden ebenfalls die syntaktischen Komponenten der Sätze den semantischen Komponenten zugeordnet²⁵. Dabei wird dem *neutralen Element* die Bezeichnung $e \in G$ zugeordnet. Da wie oben beschrieben eine solche einfache mathematische Zeichenkette analysierbar ist, kann dies übersetzt werden in: *e [neutrales Element] ist ein Element von G [Gruppe]*. Daher kann das neutrale Element als Konzept mit einer *is_a*-Beziehung direkt an das Konzept GRUPPE gebunden werden:

[*is_element_of*, Gruppe, neutrales Element]

Das neutrale Element weist weitere Eigenschaften auf. Die erste Eigenschaft ist nicht weiter auflösbar, da sie im Wesentlichen nur aus einer Formel besteht, die an dieser Stelle noch nicht analysierbar ist. Aus der zweiten Eigenschaft können weitere Informationen gewonnen werden. Es existiert wiederum die definierende Klammer „*inverses Element von a genannt*“ und die Beschreibung „*a' ist ein Element von G*“.²⁶ Es sind damit zwei mögliche Verbindungen konstruierbar.

- [*is_element_of*, Gruppe, inverses Element]
- [*is_property*, neutrales Element, inverses Element]

Hierbei tritt wiederum ein Problem auf. Der gegenwärtige Analyseprozess erkennt den Zusammenhang zwischen neutralem und inversem Element nicht vollständig.

Diese einzelnen semantischen Informationen lassen sich in die Wissensbasis integrieren, indem drei neue Konzepte gebildet werden: GRUPPE, INVERSES ELEMENT und NEUTRALES ELEMENT. Die Wissensbasis muss dazu die Konzepte

²⁵Da die Sätze relativ einfach sind, soll die Zuordnung von syntaktischen zu semantischen Komponenten nicht explizit gezeigt werden.

²⁶Zu den Schwierigkeiten für eine genauere Analyse dieser Sätze, siehe Kapitel 5.

MENGE, VERKNÜPFUNG und ASSOZIATIVGESETZ enthalten, wobei die Bezeichnung *Assoziativgesetz* durch das Konzept ASSOZIATIV mit der Eigenschaft, dass die Bezeichnung auch *Assoziativgesetz* sein kann, realisiert wird. Dies wird in Abbildung 3.33 dargestellt.

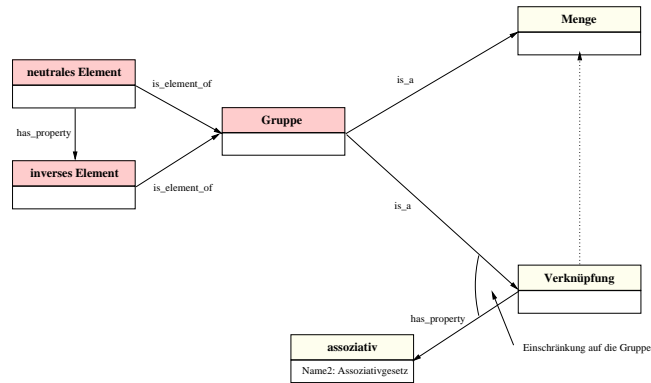


Abbildung 3.33: Darstellung in der Ontologie

Die Entität *Satz*. Sätze besitzen eine komplexere semantische Struktur, die schwieriger zu analysieren ist als die von Definitionen. Es existieren verschiedene Binnenstrukturen: Implikationen, Äquivalenzen und einfache Aussagen. Die wichtigsten semantischen Träger in den Sätzen sind die Verben und die Phrasen für Implikationen und Äquivalenzen, die zu den entsprechenden logischen Operatoren gehören. Alle Begriffe, die in einem Satz auftreten, müssen *vor* der Analyse in der Wissensbasis vorhanden sein. Es fehlen also die Beziehungen zwischen den Konzepten, die die Implikation *is_implication* und die Äquivalenz *is_equivalent* repräsentieren. Außerdem muss ein Satz anders gehandhabt werden als das semantischen Netz, das durch Definitionen gebildet wird. Eine Vermischung der Sachverhalte und Begriffe würde sonst zu Komplikationen führen. So besteht z. B. eine Äquivalenz aus zwei Aussagen. Jede Aussage wird in der Wissensbasis zusammengefasst bzw. eingekapselt (**Aussagenkapselung**). Dabei ist auch die Äquivalenzbeziehung eine Aussage, die ebenfalls gekapselt wird. Dadurch ist es möglich, komplexere Strukturen als einfache Aussagen zu bilden und diese wieder mit anderen Strukturen in Beziehung zu setzen, ohne das Begriffsnetz zu beeinflussen.

Als erstes soll ein einfacher Aussagesatz aus Beispiel 3.3.25 betrachtet werden:

Jeder Vektorraum besitzt eine Basis.

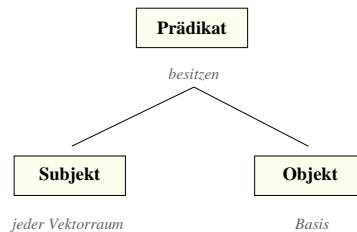


Abbildung 3.34: Zerlegung des Theorems in semantische Bausteine

Dieser Satz ist relativ einfach zu analysieren. Vorausgesetzt, dass die Begriffe Vektorraum und Basis in der Wissensbasis enthalten sind, wird durch das Prädikat eine neue Beziehung *belong* induziert. In diesem Fall wird auch der Allquantor (gekennzeichnet durch das Wort *jeder*) verwendet, der durch ein Unterkonzept zum Konzept VEKTORRAUM realisiert wird:

- [*belong*, Basis, Vektorraum_{ALL}]

Damit wird die in Abbildung 3.35 Struktur in der Wissensbasis erzeugt.

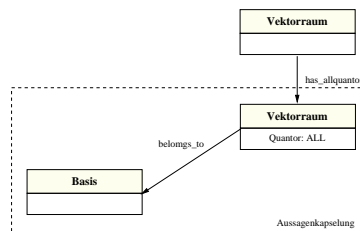


Abbildung 3.35: Darstellung in der Wissensbasis für das Beispiel 3.3.25

Exemplarisch für eine Implikation soll nun der Basisergänzungssatz 3.3.26 in der Wissensbasis dargestellt werden.

Basisauswahlsatz

Sei B eine Menge von Vektoren des Vektorraums V . B ist genau dann eine Basis, wenn B ein minimales Erzeugendensystem ist.

In diesem Fall muss zuerst die Voraussetzung „*Sei B eine Menge von Vektoren des Vektorraums V .*“ betrachtet werden. Voraussetzungen werden nicht in die Wissensbasis integriert, sondern nur temporär angelegt. Die einzige in ihnen enthaltene semantische Information ist, welchen Begriff sie einer mathematischen Zeichenkette zuordnen. Dieses Symbol wird dann als Synonym des mathematischen Begriffs innerhalb der Entität verwendet. Folgende semantischen Informationen lassen sich aus der Voraussetzung des Basisergänzungssatzes gewinnen:

- $[is_a, \text{Vektorraum}, V]$
- $[is_a, \text{Menge von Vektoren des Vektorraums}, B]$

Anschließend wird die Implikation behandelt. Der Satz wird gemäß seiner Binnenstruktur aufgelöst und in zwei Komponenten zerlegt: *B ist genau dann eine Basis* und *B ist ein minimales Erzeugendensystem*. Diese Zerlegung wird in Abbildung 3.36 dargestellt:

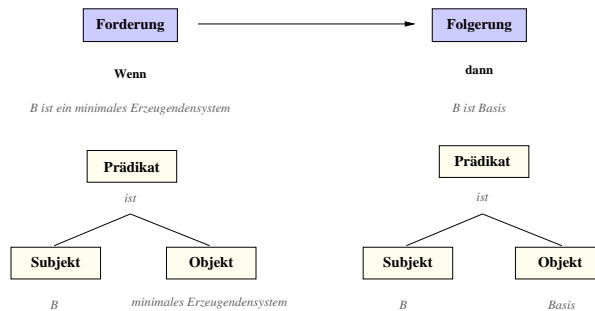


Abbildung 3.36: Zerlegung des Satzes in semantische Bausteine

Nun müssen zunächst Forderung und Folgerung intern aufgelöst werden:

- $[is_a, \text{minimal erzeugendes System}, \text{Menge von Vektoren}]_{\text{Forderung}}$
- $[is_equivalent, \text{Basis}, \text{Menge von Vektoren}]_{\text{Folgerung}}$
- $[is_implication, \text{Forderung}, \text{Folgerung}]$

In Abbildung 3.37 wird die Visualisierung des Basisergänzungssatzes in der Wissensbasis dargestellt. Die gestrichelten Bereiche deuten die Kapselung an.

Forderung, Folgerung und die gesamten Aussagen werden dabei gekapselt, wobei die *is_implication*-Beziehung in diesem Fall nur die Kapselungen verbindet. Dementsprechend ist eine Kapselung ein spezielles Konzept in der Wissensbasis.

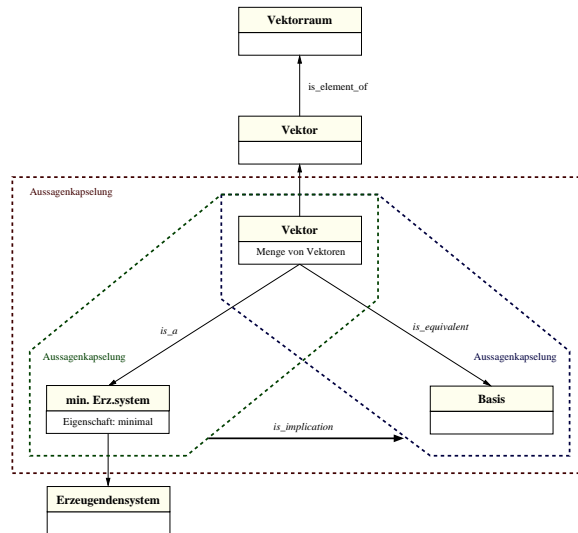


Abbildung 3.37: Darstellung in der Wissensbasis des Beispiels 3.3.26

Die Entität *Axiom*. Axiome sind unbeweisbare Tatsachen, die in Texten nur sehr selten auftreten und die daher grundsätzlich manuell in die Wissensbasis integriert werden. Damit soll verhindert werden, dass die Integrität der Ontologie gefährdet wird.

3.4.3 Zusammenfassung

Die Grundstruktur der Wissensbasis besteht aus der axiomatischen Mengenlehre und der Prädikatenlogik. Mithilfe dieser Theorien werden grundlegende mathematische Strukturen in einer Ontologie als Konzepte und Beziehungen zusammengestellt. Begriffe, die durch Definitionen eingeführt werden, sind Konzepte in der Wissensbasis. Der Konzeptname wird durch den Begriffsnamen festgelegt. Die Konzepte werden dann über *is_a*-Beziehungen mit anderen Konzepten

verbunden. Eigenschaften werden mittels der *has_property*-Beziehung an das betreffende Konzept gebunden. In diesem Prozess können aber nur Begriffe bzw. Konzepte aufgenommen werden, die sich auf Konzepte beziehen, die schon in der Wissensbasis vorhanden sind.

Sachverhalte werden durch Sätze beschrieben. Die Integration von Sätzen ist ungleich schwerer als die von Definitionen. Es werden dabei keine neuen Konzepte eingeführt, sondern Sachverhalte miteinander verbunden. Dadurch entstehen komplexe Strukturen in der Wissensbasis. Es werden dabei beispielsweise über Äquivalenzbeziehungen Aussagen miteinander verknüpft. Um das Gesamtgefüge eines Satzes nicht zu zerstören, werden die Aussagen gekapselt. Für die einzelnen Satztypen existieren verschiedene Beziehungen, die sich aus der Binnenstruktur ableiten lassen: *is_implication* und *is_equivalent*. Für einfache Aussagen ist das Prädikat der Träger der semantischen Information. Daher ist das Verb für die Bildung von Beziehungen verantwortlich.

Kapitel 4

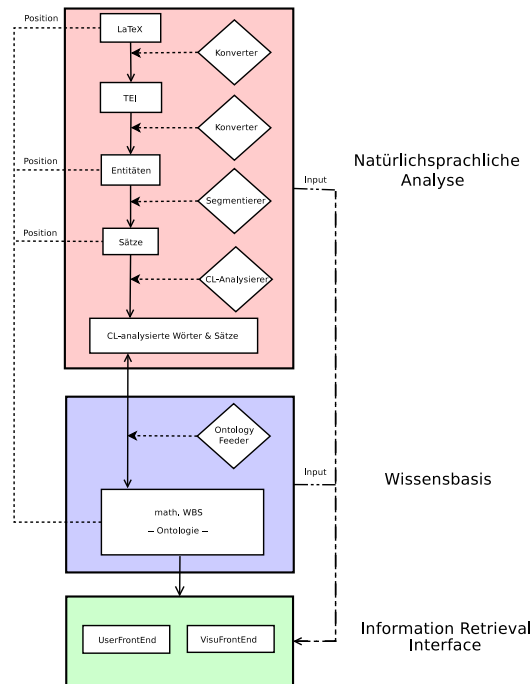
Architektur von mArachna

*Das Unsympathische an Computern ist, dass sie
nur ja oder nein sagen können, aber nicht vielleicht.*
(Brigitte Bardot)

4.1 Gesamtkonzeption

In Kapitel 1.2 wurden Hypothesen aufgestellt, die nun durch die Implementierung eines Prototypen in Software überprüft werden sollen. Dieser Prototyp — *mArachna* — wird Entitäten (Kapitel 3.3.1) semantisch analysieren und die enthaltenen semantischen Informationen einer mathematischen Wissensbasis zur Verfügung stellen.

mArachna besteht im Wesentlichen aus drei Komponenten (Abbildung 4.1): der **natürlichsprachlichen Analyse**, der **Wissensbasis** und dem **Information Retrieval System**. Die Wissensbasis stellt das Herzstück von *mArachna* dar: Sie verwaltet mathematische Informationen, unterstützt den semantischen Anteil der natürlichsprachlichen Analyse und präpariert die mathematischen Informationen für das Information Retrieval System. Die mathematischen Informationen generiert *mArachna* aus den eingelesenen mathematischen Texten, die aus Entitäten bestehen. Entitäten sind nach Kapitel 3.3.1 Definitionen, Theoreme, Sätze usw.

Abbildung 4.1: Architektur von *mArachna*

Dabei können die Eingabetexte in verschiedenen Formaten vorliegen, die dann mit Hilfe eines Konverters in das XML-Format TEI (Text Encoding Initiative [TEIa]) umgewandelt werden. In weiteren Analyseschritten werden die Texte zerlegt, bis nur noch einzelne Wörter vorliegen. Dabei bleibt aber die Position der einzelnen Wörter im Gesamttext erhalten, so dass auch Sätze bzw. Texte komplett bearbeitet werden können. Die zerlegten Texte werden dann morphologisch, syntaktisch und semantisch analysiert. Am Ende des Prozesses werden *semantische Einheiten* für die Wissensbasis generiert. Die Übergabe der semantischen Einheiten an die Wissensbasis erfolgt dann durch den **Ontology Feeder**. Dabei sollen nur mathematische Informationen aufgenommen werden, für die bereits Voraussetzungen (mathematische Grundlagen) in der Wissensbasis existieren. Die Wissensbasis selbst ist eine Ontologie, die durch einen RDF-[RDF] und einen OWL-Parser [OWL] erzeugt wird.

Das Information Retrieval System unterteilt sich in zwei Komponenten: das klassische *Benutzerfrontend* (**UserFrontEnd**) und das *Visualisierungsfrend*

(**VisuFrontEnd**). Das VisuFrontEnd dient zur Darstellung der einzelnen Analyseschritte von *mArachna*. Administratoren des Prototypen können hier über ein graphisches Interface einzelne Analyseschritte einsehen und diese gegebenenfalls korrigieren (*semiautomatischer Ansatz*). Das UserFrontEnd bietet dem Benutzer die Möglichkeit, Fragen an das System zu stellen und Antworten in geeigneter Form zu präsentieren. Ein mögliches UserFrontEnd ist z. B. ein *mathematisches Lexikon*.

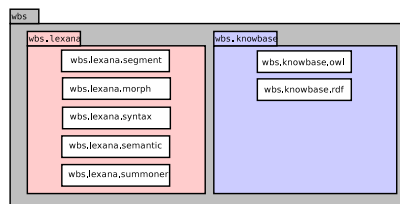


Abbildung 4.2: Pakethierarchie

Für die Implementierung von *mArachna* wurde hauptsächlich die Programmiersprache *Java* verwendet. Die Software besteht aus zahlreichen einzelnen Anwendungen, die jeweils eine bestimmte Aufgabe im Analyseprozess erfüllen. Die Gesamtstruktur wird in Abbildung 4.2 dargestellt. Die für die natürlichsprachliche Analyse relevanten Teile sind im Paket **wbs.lexana** zusammengefasst:

- **wbs.lexana.segment** (WordTokenizer)
- **wbs.lexana.morph** (morphologische Analyse)
- **wbs.lexana.syntax** (syntaktische Analyse)
- **wbs.lexana.semantic** (semantische Analyse)
- **wbs.lexana.summoner** (OntologyFeeder)

Die Wissensbasis ist im Paket **wbs.knowledge** enthalten:

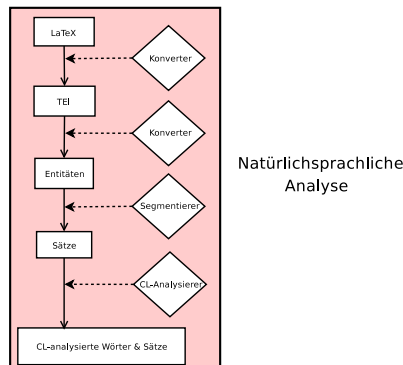
- **wbs.knowledge.rdf** (RDF-Parser)
- **wbs.knowledge.owl** (OWL-Parser)

Das Information Retrieval System ist eine webbasierte Benutzeroberfläche (*WebFrontEnd*), die die Ausgaben von *mArachna* verarbeitet und darstellt.

In den folgenden Abschnitten werden die drei Komponenten natürlichsprachliche Analyse, Wissensbasis und Information Retrieval System genauer dargestellt.

4.2 Natürlichsprachliche Analyse

*Worte sind Gegenstände:
Man kann sich an ihnen stoßen.*
(Wolfgang Herbst)



Die natürlichsprachliche Analyse umfasst die **morphologische**, **syntaktische** und **semantische Analyse**. Diese Analysen sind nicht eindeutig voneinander zu trennen und beeinflussen sich gegenseitig. Allerdings werden in *mArachna* zuerst die morphologischen und syntaktischen Analysen durchgeführt. Zum Schluss erfolgt auf Basis dieser beiden Analysen die semantische Analyse. Somit ist eine getrennte Betrachtung der einzelnen Komponenten möglich.

Die in der natürlichsprachlichen Analyse verwendeten Verfahren (Kapitel 2.4) sind sehr einfache und zumeist regelbasiert. Im Folgenden wird anhand eines Beispieltextes (Abbildung 4.3) die natürlichsprachliche Analyse diskutiert. Allerdings soll nicht der gesamte Text betrachtet werden, da dadurch die Analyse zu kompliziert wird, um sie auf Papier darzustellen.

4.2.1 Zerlegung von \LaTeX in Satzteile, Wörter und Formeln

Der mathematische Text wird zunächst vom \LaTeX -Format in das TEI-Format konvertiert. TEI steht dabei für *Text Encoding Initiative* [TEIa, SMB03a, SMB03b, TEIb].

Definition

Seien V eine Menge und K ein Körper mit den zwei Verknüpfungen:

- $V \times V \rightarrow V$ Addition mit $\langle a, b \rangle \mapsto a + b$.
- $K \times V \rightarrow V$ Multiplikation mit einem Skalar mit $\langle \alpha, b \rangle \mapsto \alpha b$.

V mit der Addition und Multiplikation heißt Vektorraum über K . Die Elemente aus V heißen Vektoren und die Elemente aus K heißen Skalare oder Koeffizienten.

Für die Addition gilt:

- $a + b = b + a, \forall a, b \in V$ (kommutativ)
- $a + (a + b) = (a + b) + c, \forall a, b, c \in V$ (assoziativ)
- $\exists 0 \in V$, so dass $a + 0 = a, \forall a \in V$ (neutrales Element)
- Zu jedem $a \in V \exists -a \in V$, so dass $a + (-a) = 0$ (inverses Element zu a)

Für die Multiplikation mit Skalaren gilt:

- $\alpha(\beta a) = (\alpha\beta)a, \forall a \in V$ und $\alpha, \beta \in K$ (assoziativ)
- $1a = a, \forall a \in V$ (neutrales Element)

Zwischen Addition und Multiplikation mit Skalaren gelten die Distributivgesetze:

- $\alpha(a + b) = \alpha a + \alpha b, \forall a \in V$ und $\alpha \in K$
- $(\alpha + \beta)a = \alpha a + \beta a, \forall a \in V$ und $\alpha, \beta \in K$

Abbildung 4.3: Beispielttext — Definition für den Vektorraum [SJZ04]

„TEI is an international and interdisciplinary standard that helps libraries, museums, publishers, and individual scholars represent all kinds of literary and linguistic texts for online research and teaching, using an encoding scheme that is maximally expressive and minimally obsolescent.“ [TEIa]

Dieses Format wurde entwickelt, um ein standardisiertes Verfahren bereitzustellen, das auch semantische Strukturen (z. B. Überschriften) und typographische Informationen von Texten in elektronischen Medien kodieren kann und dabei unabhängig vom verwendeten Betriebssystem ist. Ursprünglich wurde die *Standard Generalized Markup Language* (SGML) verwendet, mittlerweile gibt es aber auch Versionen in XML.

„Interessant sind die TEI-Richtlinien für jeden, der einen elektronischen Text in einem dauerhaften Format speichern will. Insbesondere Textwissenschaftler aller Art, z. B. Literaturwissenschaftler oder Linguisten, werden in TEI eine Möglichkeit finden, sich von proprietären Systemen der Textspeicherung (wie z. B. MS-Word) oder von offenen Standards, deren Komplexität den gestellten Aufgaben nicht gewachsen ist (z. B. HTML), frei zu machen.“ [TEIb]

Das TEI-Format bietet sich daher als Standardschnittstelle für die natürlichsprachliche Analyse an. Dadurch existiert eine Schnittstelle für weitere Parser, die z. B. mathematische Texte von HTML nach TEI übersetzen können. TEI ist aufgrund der Verwendung von XML sehr flexibel, so dass auch mathematische Inhalte mittels MathML 2.0 ([W3C04], siehe unten) integriert werden können. Aus den genannten Gründen wurde in *mArachna* die XML-Variante TEI P4 verwendet.

„MathML is an XML application for describing mathematical notation and capturing both its structure and content. The goal of MathML is to enable mathematics to be served, received, and processed on the World Wide Web, just as HTML has enabled this functionality for text.“ [W3C03]

Mit MathML können also nicht nur mathematische Zeichen in Webseiten dargestellt (*Presentation Markup*), sondern auch mathematische Inhalte semantisch

ausgezeichnet werden (*Content Markup*). Die Interpretation von semantischen Inhalten wird dann erforderlich, wenn z. B. verschiedene Notationen verwendet werden. Somit bietet MathML die Möglichkeit, mathematische Formeln zu interpretieren und sie somit für die semantische Analyse von mathematischen Texten zugänglich zu machen. In Abbildung 4.4 ist ein Ausschnitt aus dem Beispieltext 4.3 im TEI-Format mit integriertem MathML zu sehen. Der gesamte Text ist in Anhang 6.1 angefügt.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TEI.2 PUBLIC "-//TEI P4/DTD Main Document Type//EN" "http://www.tei-c.org/Guidelines/DTD/tei2.dtd"
[<!ENTITY % TEI.XML 'INCLUDE' ><!ENTITY % TEI.prose 'INCLUDE' >]>
<TEI.2 xml:lang="de">
<teiHeader>
[... ]
</teiHeader>
<text>
<body>
<div0>
<div1 type="definition">
<p>Seien  $</math> eine Menge,  $</math> ein Körper mit den zwei Verknüpfungen: </p>
<list type="bulleted">
<item><math display="http://www.w3.org/1998/Math/MathML" display="inline">
<math>\langle \mathbb{V} \rangle \times \langle \mathbb{V} \rangle \rightarrow \langle \mathbb{V} \rangle$ 
</math> Addition mit  $</math> Multiplikation mit einem Skalar mit  $</math>. </item>
</list>
<p><math display="http://www.w3.org/1998/Math/MathML" display="inline">$$$ 
```

Abbildung 4.4: TEI-MathML-Textausschnitt

Nach der Konvertierung des \LaTeX -Textes in das TEI-Format wird der Text zunächst in einzelne Entitäten zerlegt. Dadurch entstehen Definitionen, Sätze usw., die schon im Originaltext ausgezeichnete Strukturelemente waren. Somit kann die straffe Struktur von mathematischen Texten (Kapitel 3.3.6) ausgenutzt werden. Daraus können bereits erste semantische Schlussfolgerungen gezogen werden, die in anderen fachsprachlichen Texten normalerweise nicht so stark ausgeprägt vorliegen. Anschließend werden die Texte in immer kleiner werdende weitere Strukturelemente zerlegt. Dies geschieht über ausgezeichnete Strukturelemente wie z. B. Listen, über Sätze bis hin zu einzelnen Wörtern und Formeln. Zu den Wörtern zählen dabei auch Satzzeichen wie Punkte und Kommata.

Diese strukturelle Zerlegung (**WordTokenizer**) erfolgt in *mArachna* durch die Klasse `wbs.lexana.segment.XMLReader`. Der XMLReader liest den in TEI kodierten Text ein und zerlegt ihn in seine Strukturelemente. Aufbauend auf

der Basisklasse `wbs.lexana.segement.structure.Structure` existieren fünf grundlegende Strukturelemente, deren Strukturhierarchie in Abbildung 4.5 dargestellt ist:

- *Environment*: gesamtes Dokument, Entitäten und Listeneinträge
- *ItemList*: Listen
- *MathEnvironment*: mathematische Elemente wie z. B. Formeln
- *Sentence*: Sätze
- *Word*: Wörter

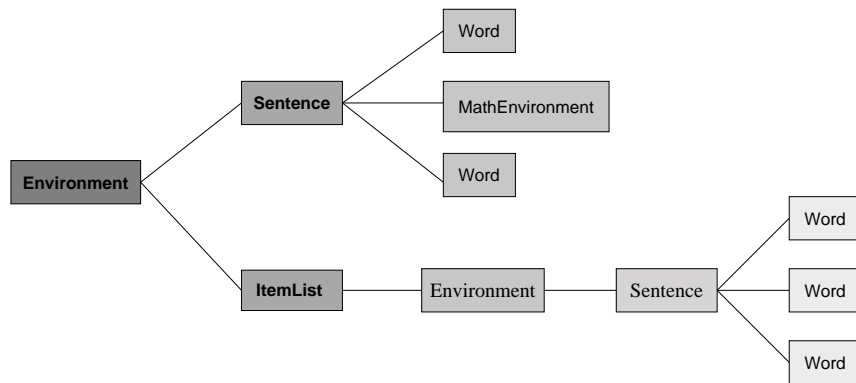


Abbildung 4.5: Textstruktur

In Beispiel 4.2.1 wird vereinfacht die Zerlegung für den Beispieltext vorgestellt¹. Das gesamte Dokument, die Entität (Definition) und die Listeneinträge (z. B.: „ $V \times V \longrightarrow V$ mit Addition [...]“) gehören der Strukturelementklasse *Environment* an, jedoch besitzen sie unterschiedliche Typen (*document*, *definition* und *item*). *Environments* werden gekennzeichnet durch:

```

<environment type="TYPNAME">
  Inhalt des Environments
</environment>

```

¹Die vollständige Datei befindet sich in Anhang 6.2

Der Tag `<item>` kennzeichnet Listen bzw. Auflistungen, die ihrerseits die oben genannten Listeneinträge enthalten. Listen gehören der Klassen `wbs.lexana.segment.structure.ItemList` an. Listen werden gekennzeichnet durch:

```
<list>
  <enviroment type="item">LISTENEINTRAG 1</environment>
  <enviroment type="item">LISTENEINTRAG 2</environment>
</list>
```

Sätze werden durch die Klasse `wbs.lexana.segment.structure.Sentence` und Wörter durch `wbs.lexana.segment.structure.Word` gekennzeichnet:

```
<sentence>
  <word>WORTEINTRAG</word>
  <word>WORTEINTRAG</word>
</sentence>
```

Als gesonderte Elemente werden mathematische Zeichen, wie Formeln und Symbole, behandelt. Sie werden durch MathML-Code integriert.

Beispiel 4.2.1 (Zerlegung des Beispieltextes 4.3 in seine Strukturelemente)

```
<environment type="document">
<environment type="definition">
  <sentence>
    <word>Seien</word>
    <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline">
      <mrow><mi>V</mi></mrow>
    </math>
    <word>eine</word>
    <word>Menge</word>
    <word>,</word>
    [...]
  </sentence>
<list>
  <environment type="item">
    <sentence>
      [...]
```



```

    </sentence>
  </environment>
  <environment type="item">
    <sentence>
      [...]
    </sentence>
  </environment>
</list>
  <sentence>
    [...]
  </sentence>
</environment>
</environment>

```

Nach dieser Zerlegung werden die Sätze als Aneinanderreihungen von Wörtern in Haupt- und Nebensätze zerlegt:

- `wbs.lexana.segment.sentence.MainClause` (Hauptsatz)
- `wbs.lexana.segment.sentence.SubClause` (Nebensatz)

Dies geschieht in `wbs.lexana.segment.structure.Sentence` mit der Methode `getSentenceType()` (Abbildung 4.6). In der Methode wird über die einzelnen Wörter iteriert und nach Kommata und Schlüsselwörtern (unterordnende Konjunktion: wenn, dass usw.) gesucht. Die Schlüsselwörter können entweder am Satzanfang (Abb. 4.6 Kasten links) oder hinter einem Komma (Abb. 4.6 Kasten rechts) stehen.

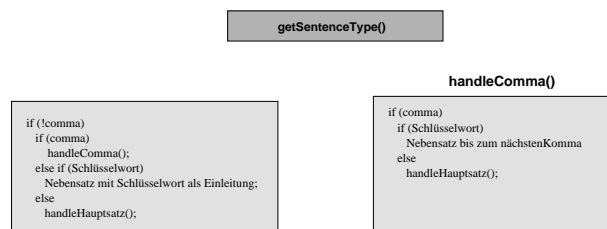


Abbildung 4.6: Funktionsweise von `getSentenceType()`

Diese Form der Verarbeitung von Sätzen ist sicherlich nicht vollständig, da u. a. uneingeleitete Nebensätze (ohne Konjunktion) und Pronominalsätze nicht als solche und als Hauptsätze gekennzeichnet werden. Jedoch werden durch die vorhandene Konstruktion viele mathematische Sätze korrekt verarbeitet. Außerdem können auf diese Art nichtgrammatikalische Konstruktionen, wie Sätze ohne Verb oder Sätze, die nur aus mathematischen Zeichen bestehen, als Hauptsatz gekennzeichnet werden.

4.2.2 Morphologische Analyse

Die morphologische Analyse, obwohl als Prozess unabhängig von der syntaktischen Analyse, ist so ausgelegt, dass die syntaktische Analyse möglichst problemlos auf sie aufsetzen kann. Sie basiert auf einem Wörterbuch, in dem alle in mathematischen Texten auftretenden Wörter manuell eingetragen werden. Es ist anzunehmen, dass sich die mathematische Sprache durch einen kleineren Wortschatz als andere Fachsprachen auszeichnet (Kapitel 2.4.4, Kapitel 1.2 Hypothese 1.2.3). Insbesondere ist die Menge der Wörter, die nicht direkt der mathematischen Terminologie angehören, eng begrenzt. Zu diesen Wörtern zählen z. B. die Artikel, Konjunktionen, häufig gebrauchte Verben (z. B.: sein, besitzen, existieren usw.). Der Rest der Wörter gehört der mathematischen Terminologie an, die in Definitionen charakterisiert wird. Es handelt sich hauptsächlich um Adjektive und Substantive. Beide Wörtergruppen bilden den *mathematischen Wortschatz*.

Das Wörterbuch von *mArachna* besteht aus zwei XML-Dateien:

- eine Wörterliste mit Suffixzuordnungen
`wbs/lexana/morph/verbosity.words.xml`
- eine Liste mit Suffixregeln
`wbs/lexana/morph/suffix.suffixes.xml`

Diese beiden Dateien werden über den entsprechenden Reader

- `wbs.lexana.morph.WordReader` für `words.xml`
- `wbs.lexana.morph.SuffixReader` für `suffixes.xml`

durch den `wbs.lexana.morph.WordAnalyzer` eingelesen. Die so erzeugten Wörter werden mit den einzelnen Wörtern des Textes verglichen.

Beispiele für Einträge in der Datei words.xml:

- SUBSTANTIV: *Abbildung*

```
<word base="abbildung" root="abbildung">
  <substantive genus="f" declination="IX" />
</word>
```

- REGELMÄSSIGES VERB: *beweisen*

```
<word base="beweisen" root="beweis">
  <verb conjugation="reg" type="full"/>
</word>
```

- UNREGELMÄSSIGES VERB: *sein*

```
<word base="sein" root="" >
  <verb conjugation="irr" type="full">
    <irregular person="3" [...] form="ist"/>
    <irregular person="3" [...] form="sind"/>
    <irregular person="3" [...] form="seien"/>
    <irregular person="3" [...] form="sei"/>
  </verb>
</word>
```

- PRÄPOSITION: *auf*

```
<word base="auf" root="auf">
  <preposition/>
</word>
```

Beispiele für Einträge in der Datei suffix.xml:

- FÜR SUBSTANTIVE:

```
<suffix verbosity="substantive" type="IX">
  <declination casus="n|g|d|a" numerus="s" suffix="*" />
  <declination casus="n|g|d|a" numerus="p" suffix="en" />
  <declination casus="n|g|d|a" numerus="p" suffix="n" />
</suffix>
```

- FÜR REGELMÄSSIGE VERBEN:

```
<suffix verbosity="verb" type="reg">
  <conjugation person="3" [...] suffix="t"/>
  <conjugation person="3" [...] suffix="en"/>
</suffix>
```

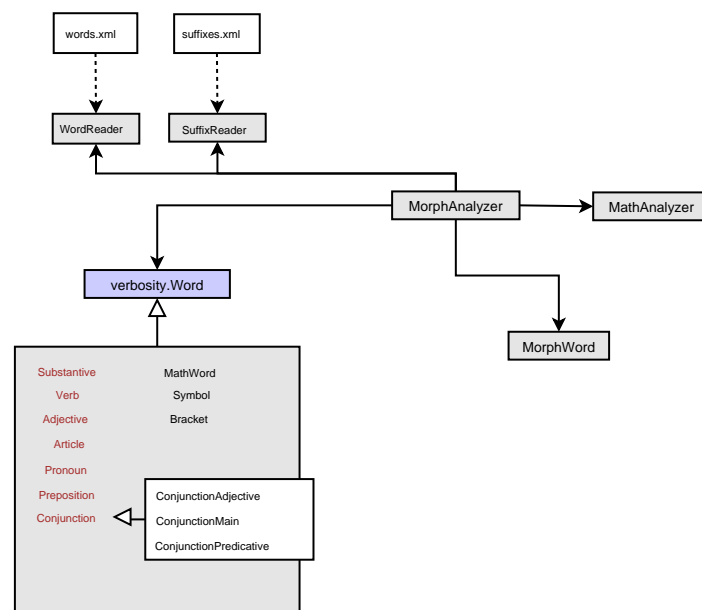


Abbildung 4.7: Struktur der semantischen Analyse

Die Wörter aus den mathematischen Texten werden den entsprechenden **Wortarten** zugeordnet, die durch Java-Klassen repräsentiert werden. Diese Klassen sind von der Basisklasse `wbs.lexana.morph.verbosity.Word` abgeleitet und entsprechen den bekannten Wortarten [Dud98]. Des Weiteren gibt es zusätzliche Wortklassen, die sich als nützlich erwiesen haben:

- Wortarten:

1. `wbs.lexana.morph.verbosity.Substantive` (Substantiv)
2. `wbs.lexana.morph.verbosity.Verb` (Verb)
3. `wbs.lexana.morph.verbosity.Adjective` (Adjektiv)
4. `wbs.lexana.morph.verbosity.Article` (Artikel)
5. `wbs.lexana.morph.verbosity.Pronoun` (Pronomen)
6. `wbs.lexana.morph.verbosity.Preposition` (Präposition)
7. `wbs.lexana.morph.verbosity.Conjunction` (Konjunktion)

- Zusätzliche Word-Klassen:

1. `wbs.lexana.morph.verbosity.MathWord`
2. `wbs.lexana.morph.verbosity.Symbol`
3. `wbs.lexana.morph.verbosity.Bracket`
4. `wbs.lexana.morph.verbosity.ConjunctionAdjective`
5. `wbs.lexana.morph.verbosity.ConjunctionMain`
6. `wbs.lexana.morph.verbosity.ConjunctionListBack`

In *mArachna* werden nicht alle Formen von Partikeln (unflektierbare Wörter) behandelt. Adverbien werden je nach ihrem Gebrauch den Adjektiven oder Präpositionen zugeordnet. Das ist zwar grammatikalisch falsch, jedoch wird die syntaktische Analyse dadurch vereinfacht. Interjektionen, eine weitere Art der Partikel, treten im mathematischen Wortschatz (Kapitel 3.3) nicht auf.

Die zusätzlichen Word-Klassen sind zur Vereinfachung der natürlichsprachlichen Analyse konstruiert worden. Die `MathWord`-Klasse beinhaltet alle mathematischen Zeichen und Formeln, die im `wbs.lexana.morph.MathAnalyzer` separat

verarbeitet werden. So können in *mArachna* einfache mathematische Symbole und Konstruktionen erkannt werden. Die **Symbol**-Klasse handhabt einzelne Buchstaben und Ziffern, die in der Mathematik hinter dem Substantiv auftreten, und wichtige Satzzeichen (z. B.: „:“, „;“, „“). Die **Bracket**-Klasse beschreibt die Klammersymbole als besondere Symbole, da in der Mathematik häufig hinter einer Formel (**MathWord**) definierende Begriffe in Klammern aufgeführt werden.

Ein zusätzliches Problem für die syntaktische Analyse sind die verschiedenen Konjunktionsverbindungen, die in der mathematischen Sprache insbesondere in den Voraussetzungen (Kapitel 3.3.2) auftreten. Die Wortart **Conjunction** wird in der morphologischen Analyse verwendet. Die letzten drei **Word**-Klassen (**ConjunctionAdjective**, **ConjunctionMain** und **ConjunctionListBack**) sind in der syntaktischen Analyse von Bedeutung. Dort wird eine **Conjunction** gegebenenfalls in eine dieser zusätzlichen Klassen umgewandelt. Sie sind grob unterteilt nach der Verwendungsart der Konjunktion (Verbindung von Wörtern, Wortgruppen oder Sätzen):

1. *Verbindung von Adjektiven*

`wbs.lexana.morph.verbosity.ConjunctionAdjective`
„f ist injektiv und surjektiv.“

2. *Hauptsatzverbindungen:*

`wbs.lexana.morph.verbosity.ConjunctionMain`
„A ist eine Menge und B ist eine Gruppe.“

3. *Aufzählungen zum Satzende²*

`wbs.lexana.morph.verbosity.ConjunctionListBack`
*„Ein Paar $(A, *)$, bestehend aus einer Menge A und einer Verknüpfung *.“*

4. *Sonstige unterordnende Konjunktionen*

`wbs.lexana.morph.verbosity.Conjunction`
„A und B seien Mengen.“

Da Entitäten verwendet werden und dort relativ strenge Regeln für die Verwendung von Wörtern gelten, werden morphologische Ambiguitäten weitestgehend

²Diese Konstruktion wurde aus rein technischen Gründen gewählt.

vermieden oder in dieser Phrase der Analyse nicht erkannt. In einer späteren Phase der morphologischen Analyse werden alle Wörter einem allgemeinen Typ bzw. einer Basisklasse `wbs.lexana.morph.MorphWord` zugeordnet. Die Instanzen dieser Basisklasse enthalten die Wortart und Zusatzinformationen, die für die semantische Analyse von Interesse sind (z. B. den Numerus).

In Beispiel 4.2.2 wird ein Ausschnitt der Protokolldatei der morphologischen Analyse des Beispieltextes 4.3 dargestellt. Es kann daraus abgelesen werden, in welchem der Zustände (Personal-, Tempus-, Kasus-, Numerus- und Genusformen) das Wort möglicherweise steht. Allerdings kann keine eindeutige Aussage über den Zustand gemacht werden. Jedoch werden in *mArachna* diese Zustände nur als Zusatzinformationen betrachtet. Ausgenommen sind die Numerusformen (Singular und Plural), die bereits verwendet werden.

Beispiel 4.2.2 (Morphologischen Analyse des Beispielsatzes 4.3)

„Seien *V* eine Menge [...]“

```
<morph>
  <word>
    <form>seien</form>
    <verb>
      <conjugation conj="full"/>
      <person person="3p"/>
      <tempus tempus="praesens"/>
      <mode mode="pre"/>
    </verb>
  </word>
  <word>
    <form>V</form>
    <math/>
  </word>
  <word>
    <form>eine</form>
    <article/>
  </word>
  <word>
    <form>menge</form>
```

```

    <substantive>
      <declination type="IX"/>
      <genus type="f"/>
      <numerus num="s"/>
      <casus casus="n|g|d|a"/>
    </substantive>
  </word>
  [...]
</morph>

```

4.2.3 Syntaktische Analyse

In der syntaktischen Analyse werden die Sätze in ihre Satzstrukturelemente zerlegt, um diese für die semantische Analyse zu präparieren. Als grundlegende Idee wurde dazu die von Chomsky (Kapitel 2.4.2) vorgeschlagene Phrasenstrukturgrammatik verwendet. Dabei sind zwei Teilschritte nötig, die für Haupt- und Nebensätze getrennt durchgeführt werden.

- eine **Vorverarbeitung**
- die **Generierung der Phrasenstruktur**

Ein Transformer (`wbs.lexana.segment.Transformer`) bereitet typische mathematische Konstruktionen anhand der Vorverarbeitungsregeln auf, die im Kapitel 3.3 aufgelistet wurden. Sie präpariert somit die einzelnen Sätze für den Phrasenstrukturgenerator. Dafür wurden eine Reihe von Regeln für typische mathematische Konstruktionen entwickelt (Kapitel 3.3), zum Beispiel:

- **Verbvorverarbeitungsregeln:**
 - Stehen die Verben *sei* oder *seien* am Satzanfang, dann verschiebe sie hinter das **MathWord**.
 „Sei *A eine Menge*“ \longrightarrow „*A* ist *eine Menge*“
 - Steht zwischen **MathWord** und Adjektiv kein Verb, so füge das Verb *sein* ein.
 „*f injektiv*.“ \longrightarrow „*f* sein *injektiv*.“

- **Konjunktionsvorverarbeitungsregeln:**

- Werden zwei Hauptsätze durch ein *und* getrennt, so bilde zwei separate Sätze.
 „Sei A eine Menge und B eine Gruppe“ \longrightarrow „ A ist eine Menge, B ist eine Gruppe“

Im Rahmen der Verbvorverarbeitung werden dem Verb Zusatzinformationen beigelegt. So ist das Verb *sei* am Satzanfang ein Kennzeichen für eine Voraussetzung (Kapitel 3.3.2) und erhält diese Zusatzinformation für die semantische Analyse, auch wenn es während der Vorverarbeitung innerhalb des Satzes verschoben wird. Der Transformer enthält dafür eine Zusammenstellung von typischen Konstruktionen der mathematischen Sprache.

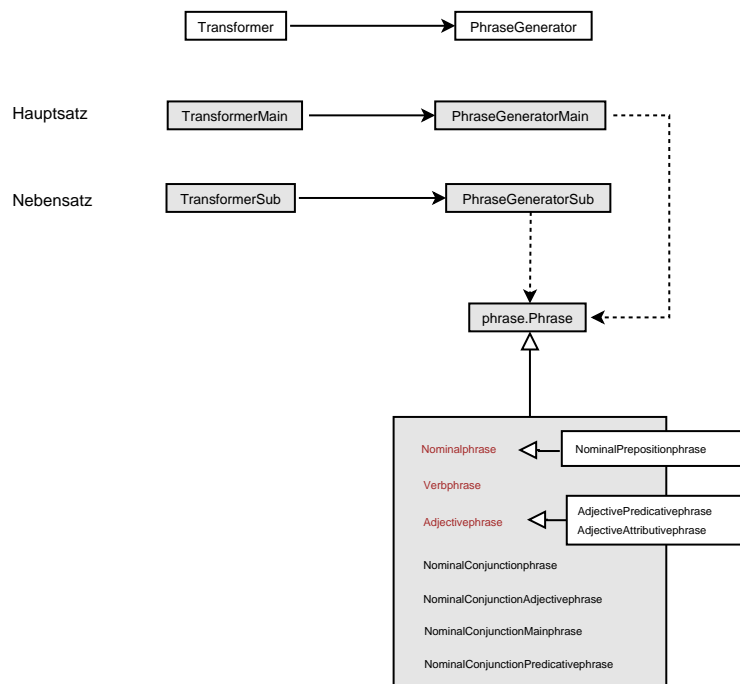


Abbildung 4.8: Struktur der syntaktischen Analyse

Nach der Vorverarbeitung werden die Phrasen gebildet. Neben den durch Chomsky bekannten Phrasen (Kapitel 2.4.2) werden weitere Phrasentypen verwendet, um die semantische Analyse zu vereinfachen:

1. Bekannte Phrase:

- **Nominalphrase**
- **Verbalphrase**
- **Adjectivephrase**

2. Zusatzphrasen:

- **NominalPrepositionphrase**
- **AdjectivePredicativephrase**
- **AdjectiveAttributivephrase**
- **NominalConjunctionphrase**
- **NominalConjunctionMainphrase**
- **NominalConjunctionPredicativephrase**

Nominalphrase

Eine grundlegende Phrase ist die Nominalphrase (NP). Sie besitzt die Komponenten bzw. Wortformen (Darstellungen in EBNF):

- $\text{Sub} ::= \text{SUBSTANTIVE} \text{ (* Substantiv *)}$
- $\text{Art} ::= \text{ARTICLE} \text{ (* Artikel*)}$
- $\text{Pro} ::= \text{PRONOUN} \text{ (* Pronomen *)}$
- $\text{Sym} ::= \text{SYMBOL} \text{ (* Symbol *)}$
- $\text{Mw} ::= \text{MATHWORD} \text{ (* mathematische Zeichen *)}$
- $\text{Pre} ::= \text{PREPOSITON} \text{ (* Präposition *)}$

Eine Nominalphrase wird im **Phrasegenerator** durch die folgenden Regeln erzeugt:

- $\text{NP} ::= [\text{Pre}] [\text{Art} \mid \text{Pro}] \text{Sub} [\text{Mw} \mid \text{Sym}];$
- $\text{NP} ::= [\text{Pre}] [\text{Art} \mid \text{Pro}] \text{Mw};$

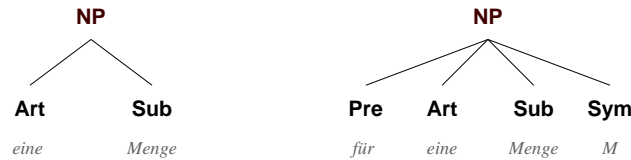


Abbildung 4.9: Beispiele für die Nominal-Phrase

Adjectivephrase

Die `wbs.lexana.segment.phrase.Adjectivephrase` ist die Basisklasse für die Verwendung von Adjektiven.

- $\text{AP} ::= (\text{Adj})+;$

Es gibt eine weitere Klasse, die von der `Adjectivephrase`-Klasse abgeleitet wird:

1. **AdjectiveAttributivephrase:** Adjektive stehen vor dem Substantiv bzw. `MathWord`

- $\text{AAP} ::= [\text{Pre}] [\text{Art} \mid \text{Pro}] (\text{Adj})+ \text{Sub} [\text{Mw} \mid \text{Sym}];$
- $\text{AAP} ::= [\text{Pre}] [\text{Art} \mid \text{Pro}] (\text{Adj})+ \text{Mw};$

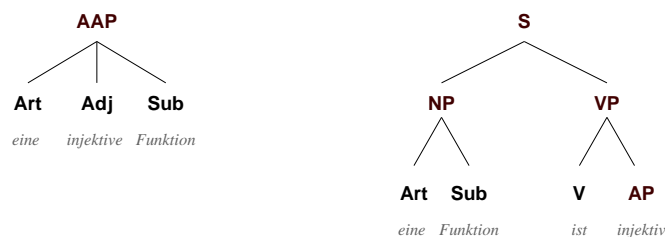


Abbildung 4.10: Beispiele für die Adjective-Phrasen

NominalConjunction-Konstruktionen

Die NominalConjunction-Konstruktionen sind *und*- oder *oder*-Verbindungen von Satzteilen, die in der Mathematik sehr häufig verwendet werden. Aufgrund der schon diskutierten Verwendung von verschiedenen Konjunktionen in der morphologischen Analyse sind zwangsläufig auch verschiedene syntaktische Phrasen notwendig.

1. **NominalConjunctionMain**: Hauptsätze, die durch eine Konjunktion getrennt sind

- $\text{NCMP} ::= \text{P P Con};$
- $\text{P} ::= \text{NP VP};$
- $\text{Con} ::= \text{CONJUNCTION};$

2. **NominalCojunctionPredicativephrase**: Aufzählungen

- $\text{NCPP} ::= \text{P1 P2 Con};$
- $\text{P1} ::= \text{NP VP};$
- $\text{P2} ::= \text{NP} \mid \text{NPP};$
- $\text{Con} ::= \text{CONJUNCTION};$

3. **NominalConjunctionAdjectivephrase**: Adjektive werden durch ein *und* getrennt

- $\text{NCAP} ::= \text{AP AP Con};$
- $\text{AP} ::= (\text{Adj})+;$
- $\text{Con} ::= \text{CONJUNCTION};$

4. **NominalConjunctionPhrase**: sonstiges

- $\text{NCP} ::= \text{P P Con};$
- $\text{P} ::= \text{NP} \mid \text{NP VP} \mid \text{NPP};$
- $\text{Con} ::= \text{CONJUNCTION};$

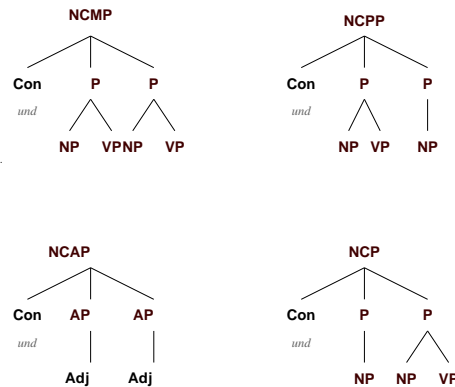


Abbildung 4.11: Beispiele für NominalConjunction-Phrasen

NominalPrepositionphrase

Die NominalPrepositionphrase ist eine Spezialkonstruktion für die Präpositional- und Konjunkionalgruppen.

1. $NPP ::= \{ NP \} \mid NP \text{ AAP}$

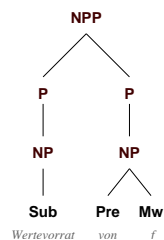


Abbildung 4.12: Beispiele für die NominalPreposition-Phrasen

Verbalphrase

Die Verbalphrase ist die zentrale Phrase. Sie besteht aus einem Verb und einer Phrase, die in den weiteren Analysen als Objekt des Satzes erkannt werden kann.

1. $VP ::= \text{Verb } P;$

2. Verb ::= VERB (* Verben*)
3. P ::= NP | NPP | NCAP | NCPP;

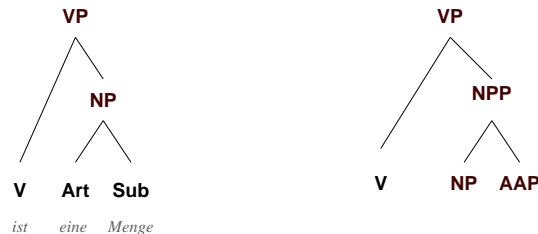


Abbildung 4.13: Beispiele für die Verb-Phrasen

Gesamtbetrachtung

Das Ergebnis ist eine Baumstruktur von Phrasen der einzelnen Sätze. In Abbildung 4.14 wird die Analyse der Voraussetzung des Beispielsatzes 4.3 verdeutlicht. Des Weiteren wird ein Ausschnitt der Protokolldatei der syntaktischen Analyse vorgestellt. Die gesamte Protokolldatei für den Beispielsatz findet sich in Anhang 6.4.

Beispiel 4.2.3 (Syntaktische Analyse des Beispieltextes 4.3)

** SATZ: Seien \$ V \$ eine Menge ,
 \$ K \$ ein Körper mit den zwei Verknüpfungen :

** HAUPTSATZ[1]: Seien \$ V \$ eine Menge

** HAUPTSATZ[2]: \$ K \$ ein Körper mit den zwei Verknüpfungen :

Nominalphrase:

MathWord: V

Verbalphrase: Verb: sein

Nominalphrase:

Artikel: eine

Substantiv: menge

Nominalphrase:

MathWord: K

Verbalphrase: Verb: sein

Nominalphrase mit Preposition:

Nominalphrase:

Artikel: ein

Substantiv: körper

Attributive Adjektiv:

Adjektiv: zwei

Preposition: mit

Artikel: den

Substantiv: verknüpfung

Symbol: :

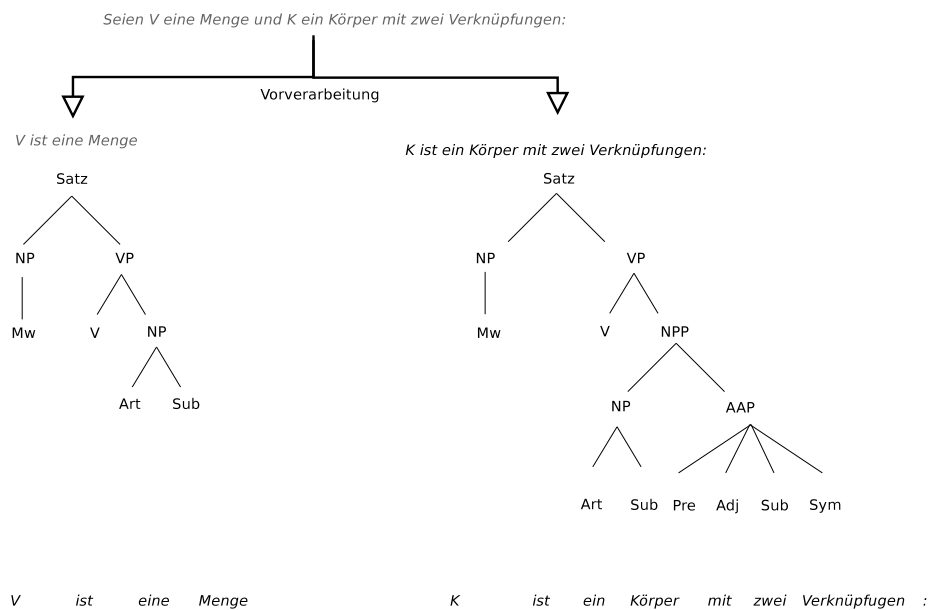


Abbildung 4.14: Analyse des Beispielsatzes

4.2.4 Semantische Analyse

Die semantische Analyse versucht unter Verwendung der Ergebnisse der morphologischen und syntaktischen Analyse die Inhalte von mathematischen Texten zu extrahieren. Dazu verwendet sie ein einfaches Modell der regelbasierten Analyse, das in drei Schichten aufgebaut ist:

1. Auflösung der Phrasenstruktur zu einer Satzbaustruktur
`wbs.lexana.semantic.SemanticPhraseAnalyzer`
2. Analyse der Satzstruktur
`wbs.lexana.semantic.SemanticPredicateAnalyzer`
3. Analyse der Entitätenstruktur
`wbs.lexana.semantic.SemanticEntityAnalyzer`

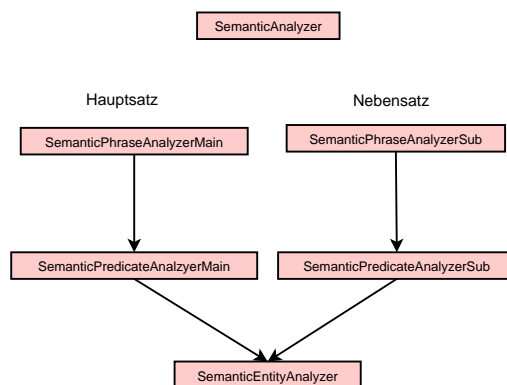


Abbildung 4.15: Semantische Analyse

Der Vorteil der mathematischen Sprache, die in den einzelnen Entitäten auftritt, ist die Einfachheit der Satzkonstruktionen. Es handelt sich um **Aussagesätze**. Die Struktur eines einfachen Satzes wird durch das Verb bestimmt. Es wird auch als **Prädikat** bezeichnet. Vom Prädikat aufgerufen stehen im Satz die **Satzglieder**. Neben dem Prädikat und den Satzgliedern existieren Verbindungselemente, die die einzelnen Sätze miteinander verknüpfen. Sätze können dadurch komplexe Strukturen bilden, die in der Mathematik aber nur selten auftreten.

4.2.4.1 Charakterisierung von Satzgliedern

In *mArachna* werden nur wenige Satzglieder verwendet. Es existieren:

- Prädikat
`wbs.lexana.semantic.phrase.Predicate`
- Subjekt
`wbs.lexana.semantic.phrase.NominalSubject`
- Objekt
`wbs.lexana.semantic.phrase.ObjectPhrase`
- Satzadjektive
`wbs.lexana.semantic.phrase.AdjectiveObject`
- Verbindungen
`wbs.lexana.semantic.phrase.NominalConjunctionSubject`

Insbesondere werden die Objekte nicht im Bezug auf ihr Genus betrachtet. Es findet daher keine Unterscheidung zwischen Akkusativobjekt, Dativobjekt und Genitivobjekt statt. Präpositionalobjekte werden über das Objekt verarbeitet. Der Satzgliedbau wird durch die Phrasenstruktur aus der syntaktischen Analyse simuliert. Die in *mArachna* getesteten Sätze aus der Beispieldefinition haben eine einfache Struktur, so dass bei der Identifizierung der Satzglieder keine Schwierigkeiten auftreten.

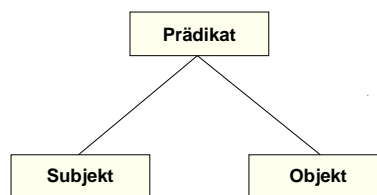


Abbildung 4.16: Baumstruktur des Satzbaus

Die Transformation der Phrasenstruktur in die Satzbaustruktur erfolgt durch den `wbs.lexana.semantic.SemanticPhraseAnalyzer` getrennt für Haupt- und Nebensätze. Folgende Regeln werden für die Analyse verwendet:

- $S ::= NP VP$;
- $NP ::= \text{Subjekt}$;
- $VP ::= V NP$ mit $V ::= \text{Prädikat}$ und $NP ::= \text{Objekt}$;

Beispiel: $A \text{ ist ein Menge.} \longrightarrow [A]_{\text{Subjekt}}[\text{ist}]_{\text{Prädikat}}[\text{eine Menge}]_{\text{Objekt}}$

4.2.4.2 Semantische Objekte

Sätze in Entitäten werden durch charakteristische Phrasen und Verben gekennzeichnet. Bei der Analyse der Satzstruktur werden diese Wortstrukturen (Kapitel 2.4.2) genutzt, um der Satzbaustruktur **semantische Objekte** zuzuordnen. Es gibt drei verschiedene Typen von semantischen Objekten für die Verarbeitung von Definitionen:

1. `wbs.lexana.semantic.semanticObject.DefinedObject`:
Charakterisiert das Element des Satzes, das definiert wird.
2. `wbs.lexana.semantic.semanticObject.DefiningObject`:
Charakterisiert die Elemente des Satzes, die das `DefinedObject` definieren.
3. `wbs.lexana.semantic.semanticObject.ConditionObject`:
Charakterisiert Nebensatzkonstruktionen, z. B. Nebensätze, die durch eine Konjunktion eingeleitet werden.

Die Transformation der Satzbaustruktur in die semantischen Objekte erfolgt durch den `wbs.lexana.semantic.SemanticPredicateAnalyzer` getrennt für Haupt- und Nebensätze.

- $\text{Prädikat} ::= \text{SubjektObjekt}$;
- $\text{Subjekt} ::= \text{DefinedObjekt}$;
- $\text{Objekt} ::= \text{DefiningObjekt}$;

Beispiel: $A \text{ ist ein Menge.} \longrightarrow$

- $[A]_{\text{Subjekt}} \text{ DefinedObjekt}$

- $[M]_{\text{Objekt}}$ **DefiningObject**

Das Prädikat bleibt erhalten und wird im nächsten Schritt semantisch verarbeitet.

4.2.4.3 Entitätenstruktur

Entitäten bestehen nicht nur aus einem grammatikalischen Satz, sondern aus einer Reihe von Sätzen, die meist in einer bestimmten Reihenfolge auftreten. So zeigen z.B. Definitionen eine typische dreigeteilte Struktur, bestehend aus **Voraussetzung**, **Aussage** und einer **Eigenschaft** oder oftmals auch einer Liste von Eigenschaften (Kapitel 3.3.2). Diese Struktur wird in Abbildung 4.17 dargestellt.

Voraussetzungen listen die vorhandenen mathematischen Objekte auf und ordnen sie gegebenenfalls mathematischen Symbolen zu. Die Aussage beschreibt das zu definierende Element und in welcher Beziehung es zu den Voraussetzungen steht. Zusätzliche Angaben werden in der Eigenschaftenliste aufgeführt. Aufgrund der logischen Struktur der Entitäten und der einfachen Konstruktion der einzelnen Sätze können semantische Informationen extrahiert werden. Phrasen, Verben und Satzbau charakterisieren die einzelnen Bausteine einer Definition. Die Erkennung von Voraussetzung, Aussage und Eigenschaften ist dadurch einfach.

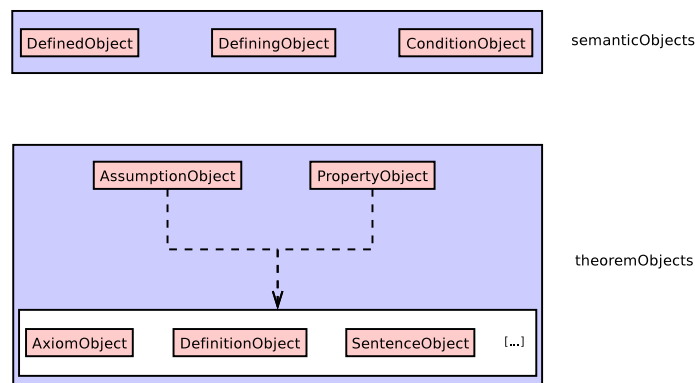


Abbildung 4.17: Entitätenstruktur

Voraussetzungen, die nach dem letzten Analyseschritt aus einer Liste von DefinedObjects und DefiningObjects bestehen, werden durch die Klasse `wbs.lexana.semantic.theoremObject.AssumptionObject` analysiert, und die extrahierten Informationen in einem speziellen Objekt gesammelt. Entsprechend werden auch Eigenschaften durch die Klasse `wbs.lexana.semantic.theoremObject.PropertyObject` verarbeitet. Beide Objekte, AssumptionObject und PropertyObject, werden der Klasse `wbs.lexana.semantic.theoremObject.DefinitionObject` zur Verfügung gestellt und regelbasiert analysiert.

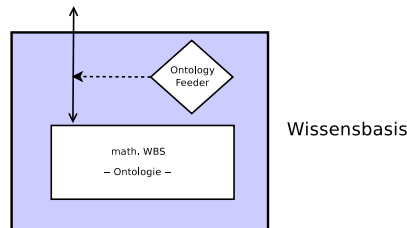
Subjekt	Prädikat	Objekt
Vektorraum	<i>is_a</i>	Menge
Vektorraum	<i>has_a</i>	Addition
Vektorraum	<i>has_a</i>	Multiplikation mit einem Skalar
Addition	<i>is_a</i>	Verknüpfung
Multiplikation mit einem Skalar	<i>is_a</i>	Verknüpfung

Abbildung 4.18: Endergebnis der semantischen Analyse des Beispieltextes

Das durch eine Liste von Tripeln (Subjekt, Prädikat, Objekt) in der Tabelle 4.18 dargestellte Ergebnis soll der Wissensbasis (Kapitel 4.3) übergeben werden. Subjekt, Prädikat und Objekt sind nicht zu verwechseln mit der Satzbauterminologie, sondern kennzeichnen hier die Teile des Ergebnistripels. Die Präpositionalphrase *Multiplikation mit einem Skalar* wird in dieser Analyse noch nicht aufgelöst (Kapitel 4.3). Dies geschieht erst im Zusammenspiel mit der Wissensbasis.

4.3 Wissensbasis

Wenn Menschen nur über das sprächen, was sie begreifen, dann würde es sehr still auf der Welt sein.
(Albert Einstein)



4.3.1 Grundkonzept

Die Wissensbasis ist das Herzstück von *mArachna*. Sie dient einerseits der Sammlung von mathematischen Informationen, andererseits unterstützt sie die semantische Analyse.

Die Wissensbasis liegt in Form einer Ontologie (Kapitel 2.2) vor, die anfangs nur eine geringe Menge im Programmcode vorgegebenen elementaren Grundlagenwissens der Mathematik enthält (Kapitel 3.4). Dabei handelt es sich um die mathematische Prädikatenlogik und die axiomatische Mengenlehre (Kapitel 3.2.2 und 3.2.3). Auf Basis dieses Grundlagenwissens wird die Ontologie um weitere mathematische Informationen erweitert. Diese Erweiterungen stammen aus den Informationen, die durch die natürlichsprachliche Analyse der mathematischen Texte extrahiert werden. Die semantischen Inhalte der einzelnen Entitäten können jedoch nicht für sich alleine stehen, da sie kontextuell gebunden sind. Die Wissensbasis dient daher als Sammelbecken für diese Informationen und stellt diese in einen kontextuellen Zusammenhang. Sie nimmt dabei keine mathematischen Informationen auf, die keinen Bezug zu einem Eintrag in der Wissensbasis haben.

Die erzeugten Wissensstrukturen (**Wissenrepräsentation**) besitzen eine komplexe Struktur (Kapitel 2.2). Einerseits werden sie durch die semantischen Strukturen der mathematischen Texte charakterisiert, andererseits bilden sie mathe-

mathematische Wissensstrukturen ab. Vorteilhaft ist dabei, dass die mathematische Sprache innerhalb der Entitäten strukturiert ist (Kapitel 3.3.2), so dass die Aussagen in diesen Entitäten mathematisches Wissen skizzenhaft schematisieren. Als Endergebnis wird eine Wissenrepräsentation erzeugt, die Teilaspekte der mathematischen Sprache und einen rudimentären Überblick mathematischen Wissens abbildet.

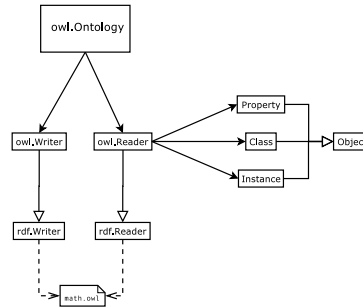


Abbildung 4.19: Überblick Wissensbasis

4.3.2 Realisierung

Für die Realisierung der Wissensbasis wird die XML-basierte Ontologiesprache OWL verwendet. Das W3C [OWL] beschreibt OWL folgendermaßen:

„OWL is a Web Ontology language. [...] OWL uses both URIs for naming and the description framework for the Web provided by RDF to add the following capabilities to ontologies:

- *Ability to be distributed across many systems.*
- *Scalability to Web needs.*
- *Compatibility with Web standards for accessibility and internationalization.*
- *Openness and extensibility.*

OWL builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes: among others, relations between

classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes.“

OWL bietet ein standardisiertes Verfahren für die computergerechte Realisierung einer Ontologie und insbesondere auch für das *Semantic Web*. Die Ontologie wird in einer XML-Datei gespeichert. Bei *mArachna* ist die formale Beschreibung der Wissensbasis in der Datei `math.owl` zu finden, die in Abbildung 4.20 als Ausschnitt abgebildet ist. Die Konzeption dieser Ontologie wurde schon

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" > ]>

<rdf:RDF
  xmlns:base="http://www.math.tu-berlin.de/~natho/mathont/"
  xmlns="http://www.math.tu-berlin.de/~natho/mathont/"
  xmlns:xsd="xsd:"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="owl:"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <owl:Ontology rdf:about="">
    <dc:Title>Mathematical Ontology</dc:Title>
    <dc:Creator>
      <rdf:Bag>
        <rdf:li>Nicole Natho</rdf:li>
        <rdf:li>Sebastian Rittau</rdf:li>
      </rdf:Bag>
    </dc:Creator>
    <dc:Publisher>Berlin University of Technology</dc:Publisher>
  </owl:Ontology>

  <!-- ===== -->
  <!-- Classes -->
  <!-- ===== -->

  <owl:Class rdf:ID="MathObject"/>
  <owl:Class rdf:ID="Classification"/>
  <owl:Class rdf:ID="Structure"/>

```

Abbildung 4.20: Ausschnitt aus der `math.owl`

ausführlich in Kapitel 3.4 beschrieben (Abbildung 3.29).

Das Bindeglied zwischen der natürlichsprachlichen Analyse und der Wissensbasis wird durch den `wbs.lexana.summoner.OntologyFeeder` dargestellt. Die semantische Analyse der mathematischen Texte (Kapitel 4.2.4) erzeugt die in Abbildung 4.18 für den Beispieltext aufgelisteten Ergebnistripel. Der OntologyFeeder überprüft, ob die verwendeten Begriffe in der Wissensbasis einen entsprechenden Eintrag besitzen. Wenn das der Fall ist, dann können die extrahierten mathematischen Informationen der Wissensbasis übergeben werden. Sind die Informationen nicht in der Wissensbasis vorhanden, dann unterbricht der OntologyFeeder den Vorgang und erzeugt eine Fehlermeldung. In einem solchen Fall muss der aufgetretene Fehler mithilfe des VisuFrontEnds (Kapitel 4.4 manuell korrigiert werden (*halbautomatischer Ansatz*). Eine solche Änderung kann auf

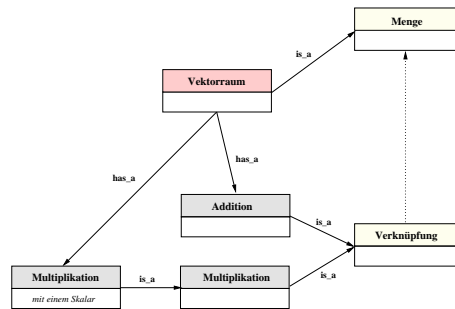


Abbildung 4.21: Visualisierung des Beispielsatzes

zwei Arten geschehen. Entweder werden weitere mathematische Texte eingelesen und somit das noch nicht vorhandene Wissen integriert, oder die Änderung der Wissensbasis erfolgt durch die direkte Änderung der Datei `math.owl`. Dies geschieht durch die entsprechenden `writer`-Klassen (Abbildung 4.19).

Analyse des Beispiels Die Analyse des Beispieltexes in Abbildung 4.3 ist relativ einfach. Durch die in der Tabelle 4.18 erstellten semantischen Informationen werden der Wissensbasis mathematische Begriffe und Sachverhalte übergeben. Die *Subjekte* und *Objekte* entsprechen den Konzepten in der Wissensbasis, die *Relationen* sind die Beziehungen zwischen den Konzepten (Kapitel 4.19) und werden in die entsprechenden Beziehungen transformiert:

- *is_a*-Beziehung \equiv *ist*-Relation
- *has_a*-Beziehung \equiv *mit*-Beziehung

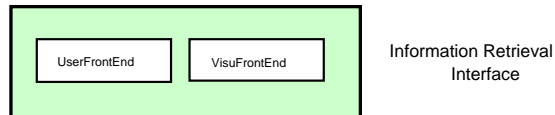
Die Objekte müssen bereits in der Wissensbasis vorhanden sein, die Subjekte können dagegen als neue Konzepte angelegt werden, müssen dies aber nicht. So werden z.B. in Sätzen üblicherweise keine neuen Begriffe eingeführt.

Problematischer ist das Subjekt *Multiplikation mit einem Skalar*. Der Begriff wird zerlegt in die Bestandteile *Multiplikation* und *mit einem Skalar*. Das Konzept MULTIPLIKATION ist schon in der Wissensbasis vorhanden. Die Präposition *mit* deutet auf eine Einschränkung hin. Daher wird ein neues Konzept mit dieser Einschränkung angelegt.

In der Abbildung 4.21 ist der Beispielsatz in der Wissensbasis visualisiert.

4.4 Information Retrieval

Am Anfang war das Wort — am Ende die Phrase
(Stanisław Jerzy Lec)



4.4.1 Grundkonzept

Das Information Retrieval System setzt sich aus zwei Komponenten zusammen:

1. Visualisierungsfrend (**VisuFrontEnd**)
2. Benutzerfrend (**UserFrontEnd**)

Das *VisuFrontEnd* ist für Administratoren und Entwickler des Prototypen konstruiert worden. In diesem FrontEnd können die einzelnen Phasen der natürlichsprachlichen Analyse und der Prozess der Wissensverarbeitung in der Wissensbasis verfolgt werden. Zusätzlich sollen in späteren Versionen von *mArachna* Schnittstellen integriert werden, in denen die Analyseschritte beeinflusst werden können.

Das *UserFrontEnd* soll die Benutzeroberfläche für Anwender von *mArachna* werden. Zur Zeit ist es nur eine prototypische Darstellung des gewünschten FrontEnds. Ziel ist die Implementierung eines interaktiven mathematischen Lexikons. Der Inhalt des Lexikons sind die Inhalte der computerlinguistisch analysierten mathematischen Texte. Das System soll auf verschiedenste Anfragen des Anwenders reagieren können, die z. B. über ein Eingabeformular an das System gerichtet werden können. Antworten des Systems können dann entweder über eine textbasierte Ausgabeschnittstelle oder über ein visualisiertes Wissensnetz erfolgen.

4.4.2 Realisierungsstand

VisuFrontEnd. Das FrontEnd ist als webbasierte Anwendung realisiert und zur Zeit noch sehr einfach gestaltet. Es werden die Protokolldateien der einzelnen Analyseschritte — \LaTeX -Parsing, Zerlegung in Satzteile, morphologische, syntaktische und semantische Analyse und Integration in die Wissensbasis — die in *mArachna* abgearbeitet werden, in graphischer Form dargestellt. Somit kann der Prozess der Analyse Schritt für Schritt verfolgt werden.

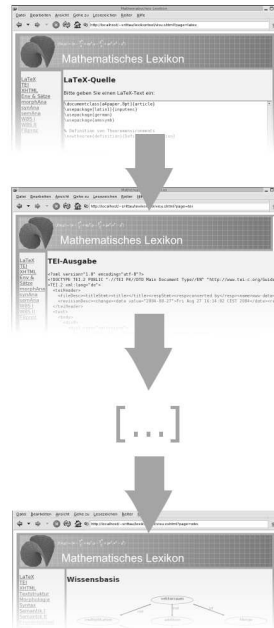


Abbildung 4.22: Analyseschritte des VisuFrontEnd

Folgende Analyseschritte werden im VisuFrontEnd dargestellt:

- **\LaTeX :** Eingabeformular für mathematische \LaTeX -Texte, die durch *mArachna* analysiert werden
- **TEI und XHTML:** Ausgabe der zu analysierenden Texte im TEI- oder XHTML-Format

- **Textstruktur:** Darstellung der Zerlegung der Texte in ihre einzelnen Strukturen: Entitäten, Sätze usw.
- **Morphologie:** Darstellung der Resultate der morphologischen Analyse der Texte
- **Syntax:** Darstellung der Zerlegung der Sätze in ihre syntaktischen Einheiten
- **Semantik I und II:** Darstellung der Resultate der einzelnen Schritte der semantischen Analyse
- **Ergebnistripel:** Darstellung des Ergebnistripels der semantischen Analyse
- **WBS:** Graphisch orientierte Darstellung des Eintrags des analysierten Textes in der Wissensbasis

UserFrontEnd. Das *UserFrontEnd* wird eine interessante Herausforderung darstellen, die über das Ziel dieser Arbeit weit hinausführen würde. Die in der Wissensbasis gesammelten mathematischen Informationen müssen wieder in geeigneter Weise dem Anwender zur Verfügung gestellt werden. Auf Anfragen des Anwenders müssen entsprechende Einträge aus der Wissensbasis extrahiert werden. Ist die Wissensbasis groß und besitzt diese viele Einträge bzw. Konzepte mit vielfältigen Beziehungen untereinander, dann ist dieser Prozess des Information Retrieval keine triviale Aufgabe.

In Abbildung 4.23 ist eine mögliche Darstellung skizziert, wie ein *UserFrontEnd* für ein mathematisches Lexikon, wie es in Kapitel 1.1 beschrieben wurde, aussehen könnte. Dabei ist die zentrale Komponente die Navigation über ein Wissensnetz, das zu einem vom Anwender angefragten Begriff oder Sachverhalt gehört. Die Knoten dieses Wissensnetzes enthalten die detaillierten Informationen über den zu betrachtenden mathematischen Begriff und ähneln den Konzepten in der Wissensbasis. Relationen entsprechen den Verbindungen zwischen den Knoten und können somit, wie die Wissensbasis, Informationen vernetzen. Daher erscheinen nach Anfragen an das System nicht nur der angefragte Begriff, sondern auch Alternativvorschläge, die mit diesem Begriff in Zusammenhang stehen. Die Darstellung der Wissensnetze kann als Text oder visuell erfolgen.

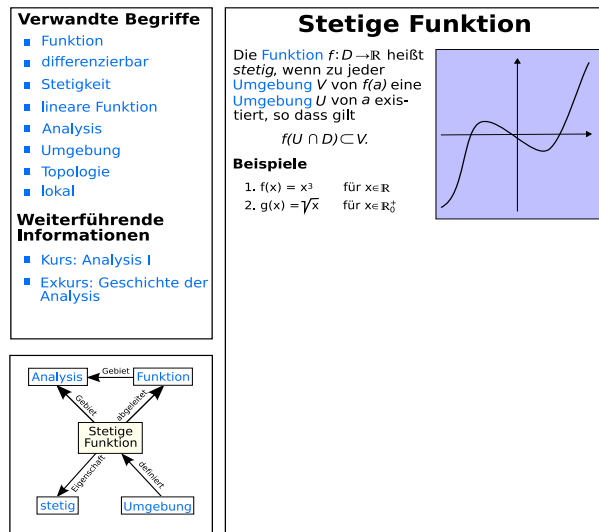


Abbildung 4.23: Das UserFrontEnd

Die Wissensnetze sind dabei kein Äquivalent der gesamten Wissensbasis, sondern nur entsprechende Abbilder von Teilbereichen dieser. Dies ist erforderlich, da die Wissensbasis ein sehr komplexes Gebilde ist. Um die Navigation sowohl für den Benutzer überschaubar als auch technisch realisierbar zu halten, ist die Entwicklung eines leistungsfähigen Navigations- und Anzeigesystems, beispielsweise auf Basis des *structure guided browsing* (Kapitel 2.5), notwendig.

Kapitel 5

Ergebnisse

5.1 Kritische Analyse

*Die Fassung der Edelsteine erhöht
ihren Preis, nicht ihren Wert.*
(Ludwig Börne)

Sprachanalyse. Die mathematische Theorie besitzt eine strenge Systematik aus gewissenhaft hergeleiteten Sätzen, die aus einem endlichen Theoremgebäude aufgebaut werden. Dadurch ist die Mathematik in kompakte Einheiten — Entitäten — zerlegbar, die miteinander in vorgegebenen Beziehungen stehen (Kapitel 3.3.2). Diese Entitäten sind Definitionen, Theoreme, Sätze, Korollare, Propositionen, Lemmata und Beweise. Definitionen sind die Strukturierungselemente einer mathematischen Theorie und die Sätze — Theoreme, einfache Sätze, Korollare, Propositionen, Lemmata — beschreiben die mathematischen Sachverhalte (Kapitel 3.3.1).

Die ausschließliche Betrachtung dieser Entitäten bei einer linguistischen Analyse ist eigentlich unzureichend. Einführende Texte zu mathematischen Themen gerade in Lehrbüchern enthalten oft Informationen, die für das weitere Verständnis eines Textes notwendig sind. Insbesondere enthalten solche Texte häufig Definitionen. Diese Texte sind jedoch schwerer zu analysieren als die klar abgegrenzten Entitäten.

Für ein vollständiges Verständnis der Mathematik sind Beweise notwendig, die in dieser Arbeit nicht betrachtet wurden. Beweise haben sehr vielfältige Konstruktionsmöglichkeiten und sind deshalb schwerer zu analysieren als die übrigen Entitäten. Darüberhinaus sind viele Beweise nur skizzenhafte Darstellungen des tatsächlichen Beweises. Allerdings können in den Beweisen wichtige Informationen enthalten sein, die für das Verständnis des weiteren Texten notwendig sind. So kann innerhalb eines Beweises ein Begriff definiert werden, der auch außerhalb des Beweises verwendet wird.

Mit Ausnahme von Beweisen und Axiomen haben mathematische Entitäten einen strikten Aufbau, bestehend aus Voraussetzung, Aussage und Eigenschaften (Kapitel 3.3.2, Abbildung 3.5). Jede auftretende Binnenstruktur besitzt ihre eigene, typische syntaktische Struktur. So zeigen z. B. die Voraussetzungen einen besonderen Aufbau, der leicht zu erkennen ist.

Bei den Definitionen existieren in den Aussagen weitere interne Binnenstrukturierungen: *Definiendum* und *Definiens*. Die Verbindung zwischen diesen beiden internen Binnenstrukturen wird unterschiedlich realisiert. So werden z. B. sehr häufig das Verb „*heißen*“ oder das mathematische Symbol „ \Leftrightarrow “ genutzt (Kapitel 3.3.3).

Die Satzinnenstruktur ist etwas komplexer, aber trotzdem überschaubar. Dabei existieren allerdings keine Unterscheidungsmerkmale zwischen den einzelnen Satztypen. Als Binnenstrukturen treten Implikationen (Kapitel 3.3.2 Abbildung 3.7), Äquivalenzen (Kapitel 3.3.2 Abbildung 3.8) und die einfache Aussage (Kapitel 3.3.2 Abbildung 3.9) auf. Die möglichen Verbindungen der internen Binnenstrukturen sind zahlreich, aber noch handhabbar. Die am häufigsten verwendeten Basiskonstruktionen sind „*[VERB1] AUSSAGE1, so/dann [VERB2] AUSSAGE2*“-Konstruktionen (Implikationen) und „*genau dann, wenn*“ (Äquivalenzen) (Kapitel 3.3.3).

Die Sätze der mathematischen Sprache besitzen eine einfache Struktur, die durch die Prädikatenlogik beeinflusst wird. Logische Operatoren und Quantoren erhalten dabei sprachliche Pendanten, die meist eindeutig zu erkennen sind (Kapitel 3.3.3). Ambiguitäten treten dabei nur in schlecht geschriebenen Texten auf, für die eine Disambiguierung erst auf semantischer Ebene oder sogar überhaupt nicht möglich wäre. Somit kann man sagen, dass die Mathematik davon deutlich weniger besitzt als die Alltagssprache (Kapitel 1.2, Hypothese 1.2.2). Durch die

Einfachheit der Struktur der Sätze ist eine syntaktische Analyse nach Chomsky (Kapitel 2.4.2) möglich. Allerdings gibt es einige Strukturen, die sich nicht so einfach durch die Phrasenstrukturgrammatik auflösen lassen. Diese müssen erst durch Vorverarbeitungsregeln in eine geeignete Form umgewandelt werden, damit sie analysiert werden können. Die Regeln, die für diese Vorverarbeitung verwendet werden, leiten sich aus typischen Phrasenkonstruktionen in der Mathematik ab (Kapitel 3.3.3). Durch die Verwendung von mathematischen Symbolen können z. B. Sätze entstehen, die keine Verben enthalten. Diese müssen dann durch eine Vorverarbeitungsregel ergänzt werden. Insgesamt ist dieser Ansatz jedoch immer noch deutlich einfacher als die für die Analyse von Alltagssprachlichen Texten verwendeten Verfahren. Damit hat sich Hypothese 1.2.1 bestätigt. Für die Analyse längerer Textpassagen wäre es trotzdem wünschenswert, andere Analyseformen einzusetzen.

Mathematische Zeichenketten bzw. Formeln haben eine eigene Struktur, die zwar ebenfalls auf der Prädikatenlogik basiert, aber nicht kompatibel zu den Strukturen der natürlichsprachlichen Texte ist. Einige mathematische Zeichenketten lassen sich in die natürliche Sprache übersetzen, wie z. B. „ \forall “ und „ $x \in X$ “. In *mArachna* werden zur Zeit nur wenige dieser Zeichenketten übersetzt (Kapitel 3.3.5, Tabelle 3.3.5). Des Weiteren werden in den Voraussetzungen häufig mathematische Zeichenketten und Symbole als Synonym für eine sprachliche Bezeichnung verwendet (Kapitel 3.3.2).

Die Struktur mathematischer Formeln ist relativ komplex und dadurch schwer zu analysieren. Jedoch ist eine semantische Analyse der Formeln wichtig für das Verständnis von mathematischen Texten. Durch die zur Zeit unzureichende Analyse der Formeln in *mArachna* gibt es Entitäten, die nicht vollständig aufgelöst werden können. So kann im Beispiel 3.3.16 das *neutrale Element* nicht vollständig erfasst werden (Kapitel 3.4). Eine Alternative wäre die zusätzliche Verwendung eines Beweissystems, um die Strukturen der Formeln zu zerlegen und zu analysieren. Dies erfordert jedoch einen ungleich größeren Aufwand.

Es gibt in der mathematischen Sprache spezielle sprachliche Konstruktionen, die von der geschriebenen Alltagssprache abweichen. So kann eine bezeichnende mathematische Zeichenkette direkt hinter der Nominalphrase stehen. Dadurch werden in *mArachna* diese Zeichen und Symbole mit in die Nominalphrase einbezogen. Des Weiteren werden die Bezeichnungen der Quantoren häufig

als Äquivalent zum Artikel gebraucht. Dadurch entstehen neue Formen der Nominalphrase:

- $NP ::= [DET/QUANT][ADJ] N [SYM]$

Darüber hinaus verhalten sich Formeln in der Nominalphrase wie Substantive, so dass das Substantiv auch durch ein so genanntes *MathWord* (MW) ersetzt werden kann.

- $NP ::= [DET/QUANT][ADJ] MW$

In der mathematischen Sprache werden häufig Präpositionalphrasen gebraucht. Die Verwendung von Präpositionen hat einen einschränkenden Charakter auf die vorherige Nominalphrase, z. B. ist ein „*Vektorraum auf dem Körper der reellen Zahlen*“ eingeschränkt auf den „*Körper der reellen Zahlen*“. Dies gilt ebenso für Adjektive. So ist der Begriff *lineare Algebra* eine auf linearen Prozessen eingeschränkte Fachdisziplin der Algebra.

Abgesehen von der großen Anzahl der explizit durch Definitionen beschriebenen Fachbegriffe besitzt die Mathematik anscheinend einen geringen Wortschatz, um Sachverhalte und Definitionen zu konstruieren. Allerdings kann nicht wie in Hypothese 1.2.3 (Kapitel 1.2) davon ausgegangen werden, dass andere Fachsprachen einen größeren Wortschatz besitzen. Aufgrund der großen Anzahl von Fachdisziplinen ist vor allem der durch die Definitionen erzeugte Wortschatz sehr groß. Zusätzlich ist anzumerken, dass die Freiheiten der Konstruktionsmöglichkeiten der deutschen Sprache von den Autoren der mathematischen Texte ausgenutzt werden. So kann ein und derselbe Begriff verschiedene sprachliche Ausprägungen besitzen (Kapitel 3.3.4).

Trotz der Strenge der mathematischen Sprache ist diese nicht unabhängig vom Sprachstil des Autors. Jeder Autor besitzt einen für ihn typischen Satz von mathematischen Phrasen, die er häufig in seinen Texten verwendet. Allerdings ist die Anzahl der möglichen Konstruktionen beschränkt, so dass die Hypothese 1.2.4 nur teilweise widerlegt ist.

Eine semantische Analyse ist ohne eine Wissensbasis nicht zu realisieren. Das Problem besteht darin, dass die semantischen Informationen einer einzelnen

Entität ohne Kontextwissen nur einen geringen Informationsgehalt besitzt. Ohne eine Möglichkeit, mathematisches Hintergrundwissen einzubringen, ist eine semantische Analyse daher sinnlos. Dieses Hintergrundwissen wird durch die Wissensbasis bereitgestellt.

Wissensbasis. Die Mathematik als Ganzes zu erfassen wird umso schwieriger, je detaillierter man diese zu beschreiben versucht. Mathematische Sachverhalte und Begriffe bilden ein großflächiges Beziehungsnetzwerk. Viele dieser Begriffe und Sachverhalte werden von den Autoren der Texte vorausgesetzt und nicht explizit in jedem Text definiert. Dabei sind die terminologischen Strukturen in der Mathematik streng und klar formalisiert, so dass jeder Begriff und Sachverhalt in der Mathematik als Ganzem genau definiert ist.

Um diese terminologischen Strukturen zu organisieren, wird in *mArachna* eine Wissensbasis verwendet, deren Grundlage eine Ontologie ist. Die konstruierte Ontologie der Mathematik besteht aus fundamentalen Erkenntnissen aus der axiomatischen Mengenlehre und der Prädikatenlogik. Begriffe, die durch Definitionen eingeführt werden, sind Konzepte in der Wissensbasis. Der Konzeptname wird durch den Begriffsnamen festgelegt. Die Konzepte sind dann über *is_a*-Beziehungen mit anderen Konzepten verbunden. Eigenschaften werden mittels der *has_property*-Beziehung an das betreffende Konzept gebunden. Die Grundidee der Wissensbasis ist es, nur Konzepte zu integrieren, die sich schon auf vorhandene Konzepte in der Wissensbasis beziehen.

Die so konstruierten Strukturen in der Wissensbasis werfen jedoch einige Fragen auf: Viele Strukturen, die in Sätzen auftreten, können nicht eindeutig beschrieben bzw. spezifiziert werden. So konnten bei der Konstruktion der Ontologie den Begriffen *geordnetes Paar*, *Abbildung* und *Relation* keine genauen kennzeichnenden Eigenschaften zugeordnet werden. Sie wurden daher durch einfache *is_a*-Beziehungen an den jeweiligen Oberbegriff gebunden (Kapitel 3.4.1). Eine adäquate Beschreibungsmethode für *geordnete Paare* sind die vorgeschlagenen Projektionsoperatoren. Allerdings kann nicht vorausgesetzt werden, dass Autoren solche Hilfskonstruktionen in ihren Texten beschreiben.

Eine elegantere Beschreibungsmöglichkeit ist die Verwendung von Hypergraphenstrukturen im semantischen Netz. Viele Sachverhalte werden in der Mathematik durch die Prädikatenlogik bestimmt. Zur Zeit ist die Ontologie der

Mathematik so aufgebaut, dass die durch die Prädikatenlogik beschriebenen Strukturen mit der sprachlichen Struktur und der Vererbungsstruktur vermischt werden. So kann z. B. das Konzept INJEKTIV in der Wissensbasis nicht eindeutig beschrieben werden. Durch die Verwendung von Hypergraphen könnten verschiedene Sprach- und Wissensebenen realisiert werden. So könnte die Definition des Konzepts INJEKTIV selbst ein kleines semantischen Netz enthalten, das die Definition prädikatenlogisch kodiert. Inwieweit diese Strukturen dann noch handhabbar in Hinsicht auf den Verarbeitungsaufwand sind, ist offen.

In der Mathematik werden Sachverhalte durch Sätze beschrieben. Allerdings ist die Integration von Sätzen in die Wissensbasis komplexer als die von Definitionen. Dabei werden keine neuen Konzepte eingeführt, sondern bekannte Konzepte miteinander verbunden. Die Beziehungen zwischen den Konzepten, die durch die Sätze gebildet werden, werden gekapselt, um diese von den übrigen Beziehungen zu trennen. Für die einzelnen Verknüpfungstypen existieren verschiedene Beziehungen: *is_implication* und *is_equivalent*. Für einfache Aussagen ist das Prädikat der Träger der semantischen Information. Daher ist das Verb für die Bildung der Beziehung verantwortlich.

Problematisch sind die Ungenauigkeiten in den mathematischen Aussagen. Insbesondere beschreiben viele Autoren Sachverhalte nur ungenau. Es werden zahlreiche Begriffe weggelassen, die für ein ungefähres Verständnis eines mathematischen Sachverhaltes nicht notwendig sind. Problematisch ist dies, wenn in einem weiteren Text der Sachverhalt genauer definiert wird und in die Wissensbasis eingebaut werden soll. Dabei können Inkonsistenzen entstehen. Eine Idee zur Vermeidung solcher Inkonsistenzen wäre die Konstruktion von verschiedenen Wissensbasen, die zunächst getrennt von einer Hauptwissensbasis aufgebaut werden. Anschließend werden die verschiedenen Wissensbasen mit der Hauptwissensbasis abgeglichen.

Trotz der beschriebenen Probleme lässt sich die Hypothese 1.2.5 bestätigen. Die vorgeschlagene Basisstruktur bietet eine gute Grundlage, um Wissen zu organisieren. Sie reicht jedoch nicht aus, um große Strukturen zu organisieren. Dafür müssen weitergehende Ansätze entwickelt werden.

Information Retrieval. Durch die Komplexität der Wissensbasis ist ein Information Retrieval System eine große Herausforderung. Auf Grund des proto-

typischen Charakters von *mArachna* können noch keine endgültigen Aussagen über ein solches System gemacht werden. Allerdings ist anzunehmen, dass es für die Wissensbasis ein ebenso komplexes Problem ist wie die Integration von semantischen Informationen in die Wissensbasis. Dies beinhaltet z. B. die Suche nach geeigneten Methoden der Navigation in der Wissensbasis (Kapitel 5.3) sowie die Entwicklung effizienter Suchmechanismen in einer großen Wissensbasis.

5.2 Zusammenfassung

*Wenn andere glauben man ist am Ende,
dann muss man erst richtig anfangen.*
(Konrad Adenauer)

Im Folgenden soll eine kurze Zusammenfassung der wichtigsten Erkenntnisse dieser Arbeit gegeben werden.

Sprachstrukturen. In dieser Arbeit wurde ein Verfahren vorgestellt, dass die mathematische Sprache computergestützt analysieren kann. Dazu wird die mathematische Sprache in ihre Bestandteile (Entitäten-, Binnenstruktur-, Satz-, Word- und Symbolstruktur) zerlegt, d. h. sie wird schematisiert. Diese Schematisierung wird von der computerlinguistischen Analyse genutzt. Eine derartige Zerlegung der mathematischen Sprache und eine darauf basierende linguistische Analyse wurde bisher nicht durchgeführt.

Eine weitere wichtige Methode ist die Reduktion auf bestimmte Textbausteine der mathematischen Sprache (Entitäten). In den Entitäten wird die Sprache der Mathematik präziser verwendet. Dadurch können wichtige Strukturen in diesen Entitäten isoliert und schematisiert werden. Komplexere Strukturen können dann sukzessiv auf diesen Grundstrukturen aufbauen.

Die Festlegung von Vorverarbeitungsregeln schafft ein Sammelbecken von Konstruktionen mathematischer Strukturen, die syntaktische mathematische Grundstrukturen erkennen können. Diese gewonnenen Grundstrukturen sind für weitere Analyseverfahren von Interesse. So sind fehlende Verben in Aufzählungen, die in dieser Arbeit vorgestellt wurden, ein Beispiel für eine solche Grundstruktur.

Wissenstrukturen. Eine wichtige Idee ist die Verknüpfung der semantischen mathematischen Sprachanalyse mit dem mathematischen Hintergrundwissen. Dieses Hintergrundwissen wird durch die Wissensbasis repräsentiert. Sie wird daher als Kontrollinstanz für die semantische Analyse genutzt.

Insbesondere verwendet die Wissensbasis als grundlegende Prinzipien die Mengenlehre und die mathematische Logik. Dies ist sicherlich nicht ausreichend für

die vollständige Beschreibung mathematischen Wissens, jedoch bilden diese beiden mathematischen Disziplinen ein solides Grundgerüst für die Strukturierung mathematischen Wissens.

mArachna. Um diese Ansätze zu überprüfen wurde ein Prototyp (mArachna) entwickelt, der die oben genannten Ergebnisse verwendet. Im mArachna-Projekt werden einzelne Entitäten, wie z. B. Definitionen und Äquivalenzen, analysiert. Die Analyse von Implikationen steht kurz vor der Fertigstellung. Analysiert wird dabei zur Zeit eine eingeschränkte Auswahl von Sätzen, die in dieser Arbeit auch als Beispiele angeführt wurden, wie z. B. die Vektorraumdefinition (Kapitel 4) oder der Basisauswahlsatz (Kapitel 3.3.3). Diese Entitäten werden dann in (Subjekt, Prädikat, Objekt)-Tripel zerlegt und z. B. für die Entität *Definition* in die Wissensbasis eingebaut. Allerdings können zur Zeit nur einfache Satzkonstruktionen verarbeitet werden. Dies ist weniger ein Problem der syntaktischen als der semantischen Analyse und lässt sich durch eine Verfeinerung der semantischen Analyse lösen, die bereits in Arbeit ist.

5.3 Ausblick

Glaube ist Gewissheit ohne Beweise.
(Henri Frédéric Amiel)

Sprachanalyse. Eine automatische Analyse der mathematischen Sprache kann nicht Aufgabe eines einzelnen Parsers sein. Die in *mArachna* verwendeten Methoden sind sehr einfach gehalten, da nur gezeigt werden sollte, dass eine Analyse der mathematischen Sprache überhaupt möglich ist. Daher erscheint es sinnvoll, auch für andere Methoden zu überprüfen, ob diese auf Grundlage der gefundenen Strukturen (Kapitel 3.3) anwendbar sind.

Zur Zeit müssen in der morphologischen Analyse neu definierte mathematische Begriffe mit ihren entsprechenden grammatikalischen Informationen manuell in ein morphologisches Lexikon eingetragen werden. Der Prozess dieser Erstellung des Lexikons ist zur Zeit sehr einfach gestaltet, jedoch recht mühselig. Daher wäre es wünschenswert, ein vorhandenes morphologisches Lexikon zu verwenden. Die Verwendung eines eigenen Lexikons hat jedoch Vorteile, so können bei der Analyse genau diejenigen Wörter gesammelt werden, die den mathematischen Wortschatz repräsentieren. In diesem Zusammenhang ist es interessant zu erfahren, wie die genaue prozentuale Verteilung der verwendeten Wörter und Phrasen in den einzelnen Entitätentypen ist. Damit ist jedoch nicht geklärt, was mit neu definierten Begriffen geschieht. Aufgrund des etwas vereinfachten Wortbildungsprozesses in der mathematischen Sprache wäre es denkbar, ein automatisiertes Verfahren zu erstellen, neu definierte Begriffe in ein morphologisches Lexikon zu integrieren.

In *mArachna* wurde die deutsche Sprache verwendet, da im Rahmen des Projektes *Mumie* deutschsprachige Studenten angesprochen werden sollen. Allerdings besitzt die englische mathematische Sprache auf den ersten Blick noch einfachere und strengere Strukturen. Da die deutsche und die englische mathematische Sprache viele Übereinstimmungen zeigen, erscheint es sinnvoll, auch englischsprachige mathematische Texte zu analysieren. Dies erscheint als sehr vielversprechend, da insbesondere viele mathematische Texte vor allem auf Englisch verfasst werden.

Des Weiteren ist die Erweiterung auf Prosatexte und Beweise eine interessante

Herausforderung, da in diesen viele Informationen enthalten sein können, die für das weitere Verständnis eines Textes notwendig sind. Dies beinhaltet aber eine komplexere linguistische Analyse.

Wissensbasis. Für die Wissensbasis stellt sich auch weiterhin die Frage, inwieweit sich die Mathematik geeignet strukturieren lässt, damit auch größere Teilgebiete der Mathematik in einer Datenstruktur erfasst werden können. Sinnvolle Ansätze dafür sind die Einteilung in verschiedene Domänen oder die Strukturbildung von Bourbaki (Kapitel 3.4). In diesem Zusammenhang bleibt zu klären, wie detailliert eine solche Darstellung gewählt werden muss, ohne die Skalierbarkeit der Wissensbasis zu gefährden.

Eine interessante Alternative wäre es, zu untersuchen, wie Menschen neu erworbenes mathematisches Wissen zu ihrem vorhandenen Wissen hinzufügen, und welche Strukturen dabei aufgebaut werden. Dies ist allerdings nicht einfach, da einerseits die zugehörigen Untersuchungen an Personen einen großen Aufwand erfordern, und andererseits bisher nicht einmal eindeutig geklärt werden konnte, wie Menschen überhaupt Wissen organisieren (Kapitel 2.2, Kapitel 2.3). Des Weiteren ist es unklar, ob die eventuell gefundenen Strukturen überhaupt mit Computern nachgebildet werden können.

Aufschlussreich wäre auch die Betrachtung der Speicherung von unvollständigem Wissen in der Wissensbasis. Speziell die Integration von verschiedenen in der Mathematik verwendeten symbolischen Notationssystemen erscheint dabei interessant.

Information Retrieval. Ein leistungsfähiges Information Retrieval System ist bis jetzt nicht in *mArachna* integriert. Dieses soll auf den in der Wissensbasis vorhandenen Informationen effiziente Suchen zu einer gegebenen Problemstellung durchführen. Eine mögliche Anwendung dafür wäre ein mathematisches Lexikon.

Dabei muss nicht nur das Skalierungsproblem auf der Wissensbasis untersucht werden, sondern auch die möglichen Formen von Anfragen eines Anwenders an ein solches System. Diese könnten z. B. als natürlichsprachliche Anfragen verwirklicht werden. Eine Möglichkeit wäre dabei, nur Anfragen zu erlauben, die mathematisch exakt gestellt werden, um die Auswertung zu vereinfachen.

Eine weitere ungeklärte Frage ist die Darstellung der Ergebnisse von Anfragen. Welche und wieviele Informationen benötigt ein Nutzer, um das gegebene Problem zu lösen bzw. die Antwort zu verstehen? In diesem Zusammenhang könnte eventuell die Verwendung von Nutzerprofilen hilfreich sein, die den z. B. Wissensstand eines Nutzers abbilden.

Kapitel 6

Anhang

6.1 TEI-Ausgabe

```
<?xml version="1.0" encoding="utf-8"?>
<TEI.2 xml:lang="de">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title/>
        <respStmt>
          <resp>converted by</resp>
          <name>www-data</name>
        </respStmt>
      </titleStmt>
      <publicationStmt>
        <authority>www-data</authority>
      </publicationStmt>
      <sourceDesc>
        <p>Original filename: unknown</p>
      </sourceDesc>
    </fileDesc>
    <revisionDesc>
```

```

<change>
  <date value="2004-10-18">
    Mon Oct 18 12:16:21 CEST 2004
  </date>
  <respStmt>
    <name>www-data</name>
    <resp>Converter</resp>
  </respStmt>
  <item>
    Automatically converted from LaTeX
    by MuLex LaTeX2TEI.
  </item>
</change>
</revisionDesc>
</teiHeader>
<text>
  <body>
    <div0>
      <div1 type="definition">
        <p>Seien
          <math xmlns="http://www.w3.org/1998/Math/MathML"
            display="inline">
              <mrow><mi>V</mi></mrow>
            </math>
          eine Menge,
          <math xmlns="http://www.w3.org/1998/Math/MathML"
            display="inline">
              <mrow><mi>K</mi></mrow>
            </math>
          ein Körper mit den zwei Verknüpfungen:
        </p>
        <list type="bulleted">
          <item>
            <math xmlns="http://www.w3.org/1998/Math/MathML"
              display="inline">

```

```

        <mrow>
          <mi>V</mi>
          <mo>&times;</mo>
          <mi>V</mi>
          <mo>&#8594;</mo>
          <mi>V</mi>
        </mrow>
      </math>
      Addition mit
      <math xmlns="http://www.w3.org/1998/Math/MathML"
        display="inline">
        <mrow>
          <mo>&lt;</mo>
          <mi>a</mi>
          <mo>,</mo>
          <mi>b</mi>
          <mo>&gt;</mo>
          <mo>&#8614;</mo>
          <mi>a</mi>
          <mo>+</mo>
          <mi>b</mi>
        </mrow>
      </math>.
    </item>
    <item>
      <math xmlns="http://www.w3.org/1998/Math/MathML"
        display="inline">
        <mrow>
          <mi>K</mi>
          <mo>&times;</mo>
          <mi>V</mi>
          <mo>&#8594;</mo>
          <mi>V</mi>
        </mrow>
      </math>

```

6.2 Zerlegung der Textstruktur

```
<?xml version="1.0"?>
```

```

<environment>
  <environment>
    <sentence>
      <word>Seien</word>
      <math xmlns="http://www.w3.org/1998/Math/MathML"
        display="inline">
        <mrow>
          <mi>V</mi>
        </mrow>
      </math>
      <word>eine</word>
      <word>Menge</word>
      <word>,</word>
      <math xmlns="http://www.w3.org/1998/Math/MathML"
        display="inline">
        <mrow>
          <mi>K</mi>
        </mrow>
      </math>
      <word>ein</word>
      <word>K&#xF6;rper</word>
      <word>mit</word>
      <word>den</word>
      <word>zwei</word>
      <word>Verkn&#xFC;pfungen</word>
      <word>:</word>
    </sentence>
  <list>
    <environment>
      <sentence>
        <math xmlns="http://www.w3.org/1998/Math/MathML"
          display="inline">
            <mrow>
              <mi>V</mi>
              <mo>&#xD7;</mo>

```

```

        <mi>V</mi>
        <mo>&#x2192;</mo>
        <mi>V</mi>
    </mrow>
</math>
<word>Addition</word>
<word>mit</word>
<math xmlns="http://www.w3.org/1998/Math/MathML"
    display="inline">
    <mrow>
        <mo>&lt;</mo>
        <mi>a</mi>
        <mo>,</mo>
        <mi>b</mi>
        <mo>&gt;</mo>
        <mo>&#x21A6;</mo>
        <mi>a</mi>
        <mo>+</mo>
        <mi>b</mi>
    </mrow>
</math>
<word>.</word>
</sentence>
</environment>
<environment>
    <sentence>
        <math xmlns="http://www.w3.org/1998/Math/MathML"
            display="inline">
            <mrow>
                <mi>K</mi>
                <mo>&#xD7;</mo>
                <mi>V</mi>
                <mo>&#x2192;</mo>
                <mi>V</mi>
            </mrow>

```

```

    </math>
    <word>Multiplikation</word>
    <word>mit</word>
    <word>einem</word>
    <word>Skalar</word>
    <word>mit</word>
    <math xmlns="http://www.w3.org/1998/Math/MathML"
      display="inline">
      <mrow>
        <mi>&#x3B1;</mi>
        <mo>,</mo>
        <mi>b</mi>
        <mo>&#x21A6;</mo>
        <mi>&#x3B1;</mi>
        <mi>b</mi>
      </mrow>
    </math>
    <word>.</word>
  </sentence>
</environment>
</list>
<sentence>
  <math xmlns="http://www.w3.org/1998/Math/MathML"
    display="inline">
    <mrow>
      <mi>V</mi>
    </mrow>
  </math>
  <word>mit</word>
  <word>der</word>
  <word>Addition</word>
  <word>und</word>
  <word>Multiplikation</word>
  <word>heißt</word>
  <word>Vektorraum</word>

```

```

<word>&#xFC;ber</word>
<math xmlns="http://www.w3.org/1998/Math/MathML"
      display="inline">
  <mrow>
    <mi>K</mi>
  </mrow>
</math>
<word>.</word>
</sentence>
</environment>
</environment>

```

6.3 Morphologische Analyse

```

<?xml version="1.0"?>
<morph>
  <word>
    <form>seien</form>
    <verb>
      <conjugation conj="full"/>
      <person person="3p"/>
      <tempus tempus="praesens"/>
      <mode mode="pre"/>
    </verb>
  </word>
  <word>
    <form> V </form>
    <math/>
  </word>
  <word>
    <form>eine</form>
    <article/>
  </word>
  <word>
    <form>menge</form>

```



```

    <substantive>
      <declination type="IX"/>
      <genus type="f"/>
      <numerus num="s"/>
      <casus casus="n|g|d|a"/>
    </substantive>
  </word>
  <word>
    <form> K </form>
    <math/>
  </word>
  <word>
    <form>ein</form>
    <article/>
  </word>
  <word>
    <form>k&#xF6;rper</form>
    <substantive>
      <declination type="II"/>
      <genus type="m"/>
      <numerus num="p"/>
      <casus casus="n|g|a"/>
    </substantive>
  </word>
  <word>
    <form>mit</form>
    <preposition/>
  </word>
  <word>
    <form>den</form>
    <article/>
  </word>
  <word>
    <form>zwei</form>
    <adjective>

```

```

    <declination type="cardinal"/>
    <numerus num="*"/>
    <casus casus="*"/>
  </adjective>
</word>
<word>
  <form>verkn&#xFC;pfungen</form>
  <substantive>
    <declination type="IX"/>
    <genus type="f"/>
    <numerus num="p"/>
    <casus casus="n|g|d|a"/>
  </substantive>
</word>
<word>
  <form>:</form>
  <symbol/>
</word>
<word>
  <form> V&#xD7;V&#x2192;V </form>
  <math/>
</word>
<word>
  <form>addition</form>
  <substantive>
    <declination type="IX"/>
    <genus type="f"/>
    <numerus num="s"/>
    <casus casus="n|g|d|a"/>
  </substantive>
</word>
<word>
  <form>mit</form>
  <preposition/>
</word>

```

```

<word>
  <form> <math>a,b\cdot a+b</math> </form>
  <math/>
</word>
<word>
  <form> K<math>\nabla V</math> </form>
  <math/>
</word>
<word>
  <form>multiplikation</form>
  <substantive>
    <declination type="IX"/>
    <genus type="f"/>
    <numerus num="s"/>
    <casus casus="n|g|d|a"/>
  </substantive>
</word>
<word>
  <form>mit</form>
  <preposition/>
</word>
<word>
  <form>einem</form>
  <article/>
</word>
<word>
  <form>skalar</form>
  <substantive>
    <declination type="II"/>
    <genus type="s"/>
    <numerus num="p"/>
    <casus casus="n|g|a"/>
  </substantive>
</word>
<word>

```

```

    <form>mit</form>
    <preposition/>
</word>
<word>
    <form> &#x3B1;,b&#x21A6;&#x3B1;b </form>
    <math/>
</word>
<word>
    <form> V </form>
    <math/>
</word>
<word>
    <form>mit</form>
    <preposition/>
</word>
<word>
    <form>der</form>
    <article/>
</word>
<word>
    <form>addition</form>
    <substantive>
        <declination type="IX"/>
        <genus type="f"/>
        <numerus num="s"/>
        <casus casus="n|g|d|a"/>
    </substantive>
</word>
<word>
    <form>und</form>
    <conjunction/>
</word>
<word>
    <form>multiplikation</form>
    <substantive>

```

```

        <declination type="IX"/>
        <genus type="f"/>
        <numerus num="s"/>
        <casus casus="n|g|d|a"/>
    </substantive>
</word>
<word>
    <form>hei&#xDF;t</form>
    <verb>
        <conjugation conj="full"/>
        <person person="nullnull"/>
        <tempus tempus="null"/>
        <mode mode="null"/>
    </verb>
</word>
<word>
    <form>vektorraum</form>
    <substantive>
        <declination type="I"/>
        <genus type="m"/>
        <numerus num="s"/>
        <casus casus="n|d|a"/>
    </substantive>
</word>
<word>
    <form>&#xFC;ber</form>
    <preposition/>
</word>
<word>
    <form> K </form>
    <math/>
</word>
</morph>

```

6.4 Syntaktische Analyse

```

<?xml version="1.0"?>
<syntax>
  <sentence>
    <text>
      Seien  $V$  eine Menge ,
       $K$  ein Körper mit den
      zwei Verknüpfungen :
    </text>
    <mainclause count="1">
      <text>
        Seien  $V$  eine Menge
      </text>
    </mainclause>
    <mainclause count="2">
      <text>
         $K$  ein Körper mit den
        zwei Verknüpfungen :
      </text>
    </mainclause>
    <analysis>
      <Nominalphrase>
        <MathWord>  $V$  </MathWord>
      </Nominalphrase>
      <verbphrase>
        <verb>sein</verb>
        <Nominalphrase>
          <article>eine</article>
          <substantive>menge</substantive>
        </Nominalphrase>
      </verbphrase>
    </analysis>
    <analysis>
      <Nominalphrase>

```

```

    <MathWord> K </MathWord>
  </Nominalphrase>
<verbphrase>
  <verb>sein</verb>
  <NominalPrepositionPhrase>
    <Nominalphrase>
      <article>ein</article>
      <substantive>k&#xF6;rper</substantive>
    </Nominalphrase>
    <AdjectiveAttributivePhrase>
      <adjective>zwei</adjective>
      <preposition>mit</preposition>
      <article>den</article>
      <substantive>verkn&#xFC;pfung</substantive>
      <symbol>:</symbol>
    </AdjectiveAttributivePhrase>
  </NominalPrepositionPhrase>
</verbphrase>
</analysis>
</sentence>
<sentence>
  <text>
    $ V&#xD7;V&#x2192;V $ Addition mit
    $ &lt;a,b&gt;&#x21A6;a+b $ .
  </text>
  <mainclause count="1">
    <text>
      $ V&#xD7;V&#x2192;V $ Addition mit
      $ &lt;a,b&gt;&#x21A6;a+b $ .
    </text>
  </mainclause>
<analysis>
  <Nominalphrase>
    <MathWord> V&#xD7;V&#x2192;V </MathWord>
  </Nominalphrase>

```

```

<verbphrase>
  <verb>sein</verb>
  <NominalPrepositionPhrase>
    <Nominalphrase>
      <substantive>addition</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>mit</preposition>
      <MathWord> <math>a,b</math><math>a+b</math> </MathWord>
    </Nominalphrase>
  </NominalPrepositionPhrase>
</verbphrase>
</analysis>
</sentence>
<sentence>
  <text>
    $  $K \in V$  Multiplikation mit
    einem Skalar mit $  $a,b \in V$  .
  </text>
  <mainclause count="1">
    <text>
      $  $K \in V$  Multiplikation mit
      einem Skalar mit $  $a,b \in V$  .
    </text>
  </mainclause>
</analysis>
  <Nominalphrase>
    <MathWord>  $K \in V$  </MathWord>
  </Nominalphrase>
  <verbphrase>
    <verb>sein</verb>
    <NominalPrepositionPhrase>
      <Nominalphrase>
        <substantive>multiplikation</substantive>
      </Nominalphrase>

```



```

    <Nominalphrase>
      <preposition>mit</preposition>
      <article>einem</article>
      <substantive>skalar</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>mit</preposition>
      <MathWord>  $\mathbb{R}, \mathbb{C}; \mathbb{R}, \mathbb{C}$  </MathWord>
    </Nominalphrase>
  </NominalPrepositionPhrase>
</verbphrase>
</analysis>
</sentence>
<sentence>
  <text>
     $V$  mit der Addition und Multiplikation
    heit Vektorraum  $V$  ber  $K$  .
  </text>
  <mainclause count="1">
    <text>
       $V$  mit der Addition und Multiplikation
      heit Vektorraum  $V$  ber  $K$  .
    </text>
  </mainclause>
<analysis>
  <NominalConjunctionphrase>
    <conjunction>und</conjunction>
    <NominalPrepositionPhrase>
      <Nominalphrase>
        <MathWord>  $V$  </MathWord>
      </Nominalphrase>
      <Nominalphrase>
        <preposition>mit</preposition>
        <article>der</article>
        <substantive>addition</substantive>

```

```

        </Nominalphrase>
    </NominalPrepositionPhrase>
    <Nominalphrase>
        <substantive>multiplikation</substantive>
    </Nominalphrase>
</NominalConjunctionphrase>
<verbphrase>
    <verb>hei&#xDF;en</verb>
    <NominalPrepositionPhrase>
        <Nominalphrase>
            <substantive>vektorraum</substantive>
        </Nominalphrase>
        <Nominalphrase>
            <preposition>&#xFC;ber</preposition>
            <MathWord> K </MathWord>
        </Nominalphrase>
    </NominalPrepositionPhrase>
</verbphrase>
</analysis>
</sentence>
</syntax>

```

6.5 Semantische Analyse

```

<?xml version="1.0"?>
<semantic>
    <predicate>sein</predicate>
    <subject>
        <NominalSubject>
            <Nominalphrase>
                <MathWord> V </MathWord>
            </Nominalphrase>
        </NominalSubject>
    </subject>
    <object>

```

```

    <Nominalphrase>
      <article>eine</article>
      <substantive>menge</substantive>
    </Nominalphrase>
  </object>
  <predicate>sein</predicate>
  <subject>
    <NominalSubject>
      <Nominalphrase>
        <MathWord> K </MathWord>
      </Nominalphrase>
    </NominalSubject>
  </subject>
  <object>
    <NominalPrepositionPhrase>
      <Nominalphrase>
        <article>ein</article>
        <substantive>k&#xF6;rper</substantive>
      </Nominalphrase>
      <AdjectiveAttributivePhrase>
        <adjective>zwei</adjective>
        <preposition>mit</preposition>
        <article>den</article>
        <substantive>verkn&#xFC;pfung</substantive>
        <symbol>:</symbol>
      </AdjectiveAttributivePhrase>
    </NominalPrepositionPhrase>
  </object>
  <predicate>sein</predicate>
  <subject>
    <NominalSubject>
      <Nominalphrase>
        <MathWord> V&#xD7;V&#x2192;V </MathWord>
      </Nominalphrase>
    </NominalSubject>
  </subject>

```

```

</subject>
<object>
  <NominalPrepositionPhrase>
    <Nominalphrase>
      <substantive>addition</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>mit</preposition>
      <MathWord> <math>a,b\in\mathbb{R}</math> </MathWord>
    </Nominalphrase>
  </NominalPrepositionPhrase>
</object>
<predicate>sein</predicate>
<subject>
  <NominalSubject>
    <Nominalphrase>
      <MathWord>  $K\cap V$  </MathWord>
    </Nominalphrase>
  </NominalSubject>
</subject>
<object>
  <NominalPrepositionPhrase>
    <Nominalphrase>
      <substantive>multiplikation</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>mit</preposition>
      <article>einem</article>
      <substantive>skalar</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>mit</preposition>
      <MathWord>  $\{a,b\}\in\mathbb{R}^2$  </MathWord>
    </Nominalphrase>
  </NominalPrepositionPhrase>

```

```

</object>
<predicate>hei&#xDF;en</predicate>
<subject>
  <NominalSubject>
    <NominalConjunctionphrase>
      <conjunction>und</conjunction>
      <NominalPrepositionPhrase>
        <Nominalphrase>
          <MathWord> V </MathWord>
        </Nominalphrase>
        <Nominalphrase>
          <preposition>mit</preposition>
          <article>der</article>
          <substantive>addition</substantive>
        </Nominalphrase>
      </NominalPrepositionPhrase>
      <Nominalphrase>
        <substantive>multiplikation</substantive>
      </Nominalphrase>
    </NominalConjunctionphrase>
  </NominalSubject>
</subject>
<object>
  <NominalPrepositionPhrase>
    <Nominalphrase>
      <substantive>vektorraum</substantive>
    </Nominalphrase>
    <Nominalphrase>
      <preposition>&#xFC;ber</preposition>
      <MathWord> K </MathWord>
    </Nominalphrase>
  </NominalPrepositionPhrase>
</object>
</semantic>

```

```

<?xml version="1.0"?>
<summoner>
  <DefinedObject>
    <MathWord> V </MathWord>
  </DefinedObject>
  <DefiningObject>
    <article>eine</article>
    <substantive count="1">menge</substantive>
  </DefiningObject>
  <DefinedObject>
    <MathWord> K </MathWord>
  </DefinedObject>
  <DefiningObject>
    <article>ein</article>
    <substantive count="1">k&#xF6;rper</substantive>
    <substantive count="2">verkn&#xFC;pfung</substantive>
    <adjective>zwei</adjective>
    <preposition>mit</preposition>
    <symbol>:</symbol>
  </DefiningObject>
  <DefinedObject>
    <MathWord> V&#xD7;V&#x2192;V </MathWord>
  </DefinedObject>
  <DefiningObject>
    <substantive count="1">addition</substantive>
  </DefiningObject>
  <DefinedObject>
    <MathWord> K&#xD7;V&#x2192;V </MathWord>
  </DefinedObject>
  <DefiningObject>
    <substantive count="1">multiplikation</substantive>
    <substantive count="2">skalar</substantive>
  </DefiningObject>
  <DefiningObject>
    <substantive count="1">multiplikation</substantive>

```

```
<conjunction>und</conjunction>
<substantive count="2">addition</substantive>
<preposition>mit</preposition>
<MathWord> V </MathWord>
</DefiningObject>
<DefinedObject>
  <substantive>vektorraum</substantive>
  <MathWord> K </MathWord>
  <preposition>&#xFC;ber</preposition>
</DefinedObject>
</summoner>
```


Literaturverzeichnis

- [And76] J. R. Anderson. *Language memory and thought*. Erlbaum, Hillsdale, 1976.
- [And83] J. R. Anderson. *The architecture of cognition*. Harvard Univ. Press, 1983.
- [And01] J. R. Anderson. *Kognitive Psychologie*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 3 edition, 2001.
- [AS77] R. Abelson and R. C. Schank. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Erlbaum, Hillsdale, 1977.
- [Aut] Automath. <http://www.cs.kun.nl/~freek/aut/>.
- [Bar94] J. D. Barrow. *Ein Himmel voller Zahlen*. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1994.
- [Bau93] P. Baumgartner. *Der Hintergrund des Wissens. Vorarbeiten zu einer Kritik der programmierbaren Vernunft*. Kärntner Druck- und Verlagsgesellschaft, Klagenfurt, 1993.
- [Bau99] J. Baur. Syntax und Semantik mathematischer Texte. Master's thesis, Universität des Saarlandes, Fachbereich Computerlinguistik, November 1999.
- [Beu94] A. Beutelspacher. *Lineare Algebra*. Vieweg, Braunschweig, Wiesbaden, 1994.

- [Beu95] A. Beutelspacher. *Das ist o.B.d.A. trivial!* Vieweg, Braunschweig, Wiesbaden, 3 edition, 1995.
- [BF71] J. D. Bransford and J. J. Franks. The abstraction of linguistic ideas. *Cognitive Psychologie*, 2:331–350, 1971.
- [BF96] M. Barner and F. Flohr. *Analysis II*. Walter de Gruyter, Berlin, New York, 3 edition, 1996.
- [BF00] M. Barner and F. Flohr. *Analysis I*. Walter de Gruyter, Berlin, New York, 5 edition, 2000.
- [BKI00] C. Beierle and G. Kern-Isberner. *Methoden wissensbasierter Systeme*. Vieweg, Braunschweig, Wiesbaden, 2000.
- [Bor77] L. Borkowski. *Formale Logik*. C. H. Beck, München, 1977.
- [Bou66] L. E. Bourne. *Human Conceptual Behavior*. Allyn & Bacon, Boston, 1966.
- [Bou74] N. Bourbaki. Die Architektur der Mathematik. In M. Otte, editor, *Mathematiker über die Mathematik*. Springer, Berlin, Heidelberg, New York, 1974.
- [Bou02] N. Bourbaki. Elemente der Mathematik. In W. Büttemeyer, editor, *Philosophie der Mathematik*. Alber, Freiburg, München, 2002.
- [Bre82] J. Bresnan, editor. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, 1982.
- [Bre00] J. Bresnan. *Lexical Functional Syntax*. Blackwell, London, 2000.
- [Bri92] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP*, pages 152–155, 1992.
- [Bri94] E. Brill. A report of recent progress in transformation-based error-driven learning. In *Proceedings of ARPA*, pages 722–727, 1994.
- [Bro03] Der Brockhaus, Computer und Informationstechnologie, 2003.

- [BS85] R. J. Brachmann and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.
- [BT81] W. F. Brewer and J. C. Treyens. Role of schemata in memory for places. *Cognitive Psychology*, 13:207–230, 1981.
- [Bur92] F. Burkowski. Retrieval Activities in a Database consisting of Heterogenous Collections of Structured Text. In *Proceedings of the 15th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 112–125, New York, 1992. ACM Press.
- [BWP97] H.-J. Bullinger, K. Wörner, and J. Prieto. *Wissensmanagement heute*. Fraunhofer Institut für Arbeitswissenschaft und Organisation, Stuttgart, 1997.
- [BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, Edinburgh, 1 edition, 1999.
- [Car98] B. Carpenter. *Type-Logical Semantics*. MIT Press, 1998.
- [CB95] A. Copestake and T. Briscoe. Semi-productive polysemy and sense extension. *Journal of Semantics*, 12:15–67, 1995.
- [CEE⁺01] K.-U. Carstensen, C. Ebert, C. Endriss, S. Jekat, R. Klabunde, and H. Langer. *Computerlinguistik und Sprachtechnologie, Eine Einführung*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2001.
- [CH77] H. H. Clark and S. E. Haviland. Comprehension and the given-new contract. In R. O. Freedle, editor, *Discourse production and comprehension*. Norwood, 1977.
- [Cho57] N. Chomsky. *Syntactic Structures*. Mouton, Den Haag, 1957.
- [Cho67] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, 1967.
- [CQ69] A. M. Collins and M. R. Quillian. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8:240–248, 1969.

- [Cru96] D.A. Cruse. *Lexical Semantics*. Cambridge University Press, 1996.
- [DAR03] DARPA. DAML - The DARPA Agent Markup Language Homepage. <http://www.daml.org/>, 2003.
- [DDF⁺90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- [DHM95] P. J. Davis, R. Hersh, and E. A. Marchisotto. *The Mathematical Experience*. Birkhäuser, Boston, Basel, Berlin, 2 edition, 1995.
- [Die75] J. Dieudonné. *Grundzüge der modernen Analysis, Band 2*. Friedr. Vieweg & Sohn mbH, Braunschweig, 1975.
- [Die00] R. Diestel. *Graphentheorie*. Springer-Lehrbuch, Berlin, Heidelberg, New York, 2 edition, 2000.
- [DtH92] M. Domenig and P. ten Hacken. *Word Manager: A System for Morphological Dictionaries*. Georg Olms Verlag AG, Hildesheim, 1992.
- [Dud98] Dudenredaktion, editor. *Duden, Grammatik der deutschen Gegenwartssprache*. Dudenverlag, Mannheim, Leipzig, Wien, Zürich, 6 edition, 1998.
- [Dud01] Dudenredaktion. *Duden, Fremdwörterbuch*. Dudenverlag, Mannheim, Leipzig, Wien, Zürich, 5 edition, 2001.
- [DWP81] D. R. Dowty, R. E. Wall, and S. Peters. *Introduction to Montague Semantics*. Reidel, Dordrecht, 1981.
- [Ebb94] H.-D. Ebbinghaus. *Einführung in die Mengenlehre*. Wissenschaftsverlag, Mannheim, 3 edition, 1994.
- [Ede96] W. Edelmann. *Lernpsychologie*. BELTZ Psychologie Verlags Union, Weinheim, 5 edition, 1996.
- [EG96] R. Evans and G. Gazdar. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22(2):167–216, 1996.

- [EJF96] H.-D. Ebbinghaus and W. Thomas J. Flum. *Einführung in die mathematische Logik*. Spektrum Akademischer Verlag, Heidelberg, 4 edition, 1996.
- [Fen04] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Berlin, Heidelberg, 2 edition, 2004.
- [Fer03] R. Ferber. *Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. dpunkt, Heidelberg, 2003.
- [fl04] Gesellschaft für Informatik. Ziele und Aufgaben der Fachgruppe Information Retrieval, 2004.
- [Fis00] G. Fischer. *Lineare Algebra*. Vieweg, Braunschweig, Wiesbaden, 12 edition, 2000.
- [fLuLdUB04] Fakultät für Linguistik und Literaturwissenschaft der Universität Bielefeld. liOn – Linguistik Online. <http://luna.lili.uni-bielefeld.de/liOn/>, 2004.
- [Fre94] G. Frege. Über Sinn und Bedeutung. *Funktion, Begriff, Bedeutung. Fünf logische Studien*, 1994.
- [Geu99] B. Geurts. *Presuppositions and Pronouns*. Elsevier Science, 1999.
- [GHL02] L. Götze and E. W. B. Hess-Lüttich. *Grammatik der deutschen Sprache*. Bertelsmann Lexikon Verlag, München, 2002.
- [GHS99] G. Grewendorf, F. Hamm, and W. Sternefeld. *Sprachliches Wissen. Eine Einführung in die moderne Theorie der grammatischen Beschreibung*. Suhrkamp, Frankfurt a. M., 1999.
- [GKPS85] G. Gazdar, E. Klein, G. K. Pullum, and I. A. Sag. *Generalized Phrase Structure Grammar*. Blackwell, Oxford, 1985.
- [GL92] A. M. Glenberg and W. E. Langston. Comprehension of illustrated text: Pictures help to build mental models. *Journal of Memory and Language*, 31:129–151, 1992.

- [GML87] A. M. Glenberg, M. Mayer, and K. Lindem. Mental models contribute to foregrounding during text comprehension. *Journal of Memory and Language*, 26:69–83, 1987.
- [GMZ97] A. C. Graesser, K. K. Millis, and R. A. Zwaan. Discourse comprehension. *Annual Review of Psychology*, 48:163–189, 1997.
- [Gö31] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. Math. Phys*, 38:173–198, 1931.
- [Gro96] QED Group. QED. <http://www.mcs.anl.gov/qed/>, 1996.
- [GRS00] G. Görz, C.-R. Rollinger, and J. Schneeberger, editors. *Handbuch der Künstliche Intelligenz*. Oldenbourg, München, Wien, 2000.
- [Gru93] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–221, 1993.
- [GS91] J. Groenendijk and M. Stokhof. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14(1):39–100, 1991.
- [Hau00] R. Hausser. *Grundlagen der Computerlinguistik*. Springer, Berlin, Heidelberg, 2000.
- [Hau02] M. Haun. *Handbuch Wissensmanagement*. Springer, Berlin, Heidelberg, 2002.
- [HB01] G. Helbig and J. Buscha. *Deutsche Grammatik. Handbuch für den Ausländerunterricht*. Langenscheidt, Berlin, München, 2001.
- [Hed02] U. Hedtstück. *Formale Sprachen und Automatentheorie*. Oldenbourg Wissenschaftsverlag GmbH, München, 2002.
- [Hei83] I. Heim. File change semantics and the familiarity theory of definiteness. In R. Bäuerle, C. Schwarze, and A. von Stechow, editors, *Meaning, Use and Interpretation of Language*. de Gruyter, Berlin, New York, 1983.
- [Hel00] H. Helbig. *Die semantische Struktur natürlicher Sprache. Wissensrepräsentation mit MultiNet*. Springer, Berlin, Heidelberg, 2000.

- [Her72] H. Hermes. *Einführung in die mathematische Logik*. Teubner, Stuttgart, 3 edition, 1972.
- [Heu93] H. Heuser. *Lehrbuch der Analysis, Teil 2*. Teubner, Stuttgart, 8 edition, 1993.
- [Hö83] H. Hörmann. *Was tun die Wörter miteinander im Satz?* Verlag für Psychologie, Göttingen, 1983.
- [Jap93] Mathematical Society Of Japan. *Encyclopedic Dictionary of Mathematics*. MIT Press, Cambridge, 2 edition, 1993.
- [JC87] M. A. Just and P. A. Carpenter. *The Psychology of Reading and Language Comprehension*. Allyn and Bacon, Boston, 1987.
- [Jes04] S. Jeschke. *Mathematik in Virtuellen Wissensräumen – IuK-Strukturen und IT-Technologien in Lehre und Forschung*. PhD thesis, Technische Universität Berlin, Berlin, April 2004.
- [JL83] P. N. Johnson-Laird. *Mental Models*. Cambridge University Press, Cambridge, 1983.
- [JL95] P. N. Johnson-Laird. Mental models, deductive reasoning, and the brain. In M. S. Gazzaniga, editor, *The Cognitive Neurosciences*, pages 999–1008. MIT Press, Cambridge, 1995.
- [JL00] P. N. Johnson-Laird. The current state of the mental model theory. In J. Garcia-Madruga, M. Carriedo, and M. J. Gonzalez-Labra, editors, *Mental Models in Reasoning*, pages 17–40. UNED, Madrid, 2000.
- [JS02] S. Jeschke and R. Seiler. Strukturen in der multimedialen Didaktik – oder: Die Mumie im Legoland (Talk). <http://www.math.tu-berlin.de/~sabina/Talks/Erlangen2002>, 2002.
- [JZ03] S. Jeschke and E. Zorn. Moses meets Mumie at Multiverse (Talk). http://www.math.tu-berlin.de/~sabina/Talks/Rhone_Island_2003, 2003.
- [Kam81] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk and M. Stokhof, editors, *Formal Methods in*

- the Study of Language*, pages 277–322. Mathematisch Centrum Tracts, Amsterdam, 1981.
- [KD78] W. Kintsch and T. A. Dijk. Towards a model of text comprehension and production. *Psychological Review*, 85:363–394, 1978.
- [Ker98] M. Kerres. *Multimediale und telemediale Lernumgebung: Konzeption und Entwicklung*. Oldenbourg, 1998.
- [KH02] L. Konieczny and B. Hemforth. Sätze und Texte verstehen und produzieren. In J. Müssler and W. Prinz, editors, *Allgemeine Psychologie*. Spektrum Akademischer Verlag, Heidelberg, 2002.
- [Kin74] W. Kintsch. *The Representation of Meaning in Memory*. Erlbaum, Hillsdale, 1974.
- [Kin88] W. Kintsch. The role of knowledge in discourse comprehension: A construction integration model. *Psychological Review*, 95:163–182, 1988.
- [Kin98] W. Kintsch. *Comprehension – A Paradigm for Cognition*. Cambridge University Press, Cambridge, 1998.
- [KK73] W. Kintsch and J. M. Keenan. Reading rate and retention as a function of the number of propositions in the base structure of sentences. *Cognitive Psychology*, 5:257–274, 1973.
- [KKP96] M. Kohlhase, S. Kuschert, and M. Pinkal. A type-theoretic semantics for λ -DRT. In P. Dekker and M. Stokhof, editors, *Proceedings of the 10th Amsterdam Colloquium*, pages 479–498, Amsterdam, 1996.
- [Kö01] K. Königsberger. *Analysis 1*. Springer, Berlin, Heidelberg, New York, 5 edition, 2001.
- [Kö04] J. Köbler. Vorlesungsskript: Theoretische Informatik II – HU Berlin. <http://fachschaft.informatik.hu-berlin.de/service/>, 2004.
- [Koe04] P. Koepke. Projekt: Darstellungsformen mathematischen Wissen. <http://www.wissensformate.uni-bonn.de/projektkoepke.htm>, 2004.

- [Kos83] K. Koskenniemi. Two-level model for morphological analysis. In A. Bundy, editor, *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 683–685, Karlsruhe, August 1983.
- [KP79] L. Karttunen and S. Peters. Conventional implicature. In C.-K. Oh and D. Dinneen, editors, *Syntax and Semantics 11, Presupposition*, pages 1–56. Academic Press, New York, 1979.
- [KvD83] W. Kintsch and T. A. van Dijk. *Strategies of Discourse Comprehension*. Academic Press, New York, 1983.
- [Lor96] F. Lorenz. *Einführung in die Algebra I*. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 3 edition, 1996.
- [MBa] The MBase Mathematical Knowledge Base Home Page. <http://www.mathweb.org/mbase/>.
- [Min74] M. Minsky. A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306*, Juni 1974.
- [MK01] A. Franke M. Kohlhasse. MBase: Representing Knowledge and Context for the Integration of Mathematical Software Systems. *Journal of Symbolic computation*, 23(4):365–402, 2001.
- [Mon74] R. Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, Yale, 1974.
- [MR92] G. McKoon and R. Ratcliff. Inference During Reading. *Psychological Review*, 99:440–466, 1992.
- [Mus96] R. Muskens. Combining montague semantics and discourse representation. *Linguistics and Philosophy*, 19:143–186, 1996.
- [NBY97] G. Navarro and R. Baeza-Yates. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.
- [NC00] S. E. Newstead and K. Coventry. The role of context und functionality in the interpretation of quantifiers. *European Journal of Cognitive Psychologie*, 12:243–259, 2000.

- [New81] A. Newell. The Knowledge Level. *The AI Magazine*, 2(2):1–20, 1981.
- [OG96] J. Oakhill and A. Garnham. *Mental models in cognitive science*. Erlbaum, Hove, 1996.
- [OMK91] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets and Systems*, 39(2):163–179, 1991.
- [Ont] Ontology Inference Layer Committee. Ontology Inference Layer. <http://www.ontoknowledge.org/oil/>.
- [Ope] OpenMath Society. OpenMath. <http://www.openmath.org/>.
- [OWL] Web Ontology Language. <http://www.w3.org/2004/OWL/>.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.
- [PF79] M. C. Potter and B. A. Faulconer. Understanding noun phrases. *Journal of Verbal Learning and Verbal Behavior*, 18:509 – 522, 1979.
- [Pin96] M. Pinkal. Radical underspecification. In P. Dekker and M. Stokhof, editors, *Proceedings of the 10th Amsterdam Colloquium*, pages 587–606, Amsterdam, 1996.
- [Pol85] M. Polanyi. *Implizites Wissen*. Suhrkamp, Frankfurt a. Main, 1985.
- [PS94] C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.
- [Qui68] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, 1968.
- [Rau79] W. Rautenberg. *Klassische und Nichtklassische Aussagenlogik*. Vieweg, Braunschweig, Wiesbaden, 1979.

- [RDF] Resource Description Framework. <http://www.w3.org/2004/RDF>.
- [Rei89] U. Reimer. *FRM: Ein Frame-Repräsentationsmodell und seine formale Semantik*. Springer-Verlag, Berlin, 1989.
- [RS99] G. Rickheit and L. Sichelschmidt. Mental models: Some answers, some questions, some suggestions. In G. Rickheit and C. Habel, editors, *Mental models in discourse processing and reasoning*, pages 9–40. North-Holland, Amsterdam, 1999.
- [RSS02] G. Rickheit, L. Sichelschmidt, and H. Strohner. *Psycholinguistik*. Stauffenburg, Tübingen, 2002.
- [Rud02] W. Rudin. *Analysis*. Oldenbourg, München, Wien, 2 edition, 2002.
- [Ryl69] G. Ryle. *Der Begriff des Geistes*. Reclam, Stuttgart, 1969.
- [Saa] Saarland Universität, TU Eindhoven, RISC Linz. OMDoc. <http://www.mathweb.org/omdoc/>.
- [Sch93] W. Schnotz. Mentale Repräsentationen beim Sprachverstehen. *Zeitschrift für Psychologie*, 201:237–259, 1993.
- [Sei00] R. Seiler. Skript zur HM 1 für Physiker. TU-Berlin, 2000. 25. Mai 1999 (überarbeitet am 3.8.00).
- [Sha92] S. C. Shapiro, editor. *Encyclopedia of Artificial Intelligence*. John Wiley and Sons, 2 edition, 1992.
- [Shi86] S. M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes, Stanford, 1986.
- [SJZ04] R. Seiler S. Jeschke, L. Oeverdieck and E. Zorn. Mumie — Multimediale Mathematikausbildung für Ingenieure. <http://www.mumie.net/>, 2004.
- [SMB03a] C. M. Sperberg-McQueen and L. Burnard, editors. *Guidelines for Electronic Text Encoding and Interchange, Volumen One*. TEI Consortium, Oxford, Providence, Charlottesville, Bergen, 2003.

- [SMB03b] C. M. Sperberg-McQueen and L. Burnard, editors. *Guidelines for Electronic Text Encoding and Interchange, Volumen Two*. TEI Consortium, Oxford, Providence, Charlottesville, Bergen, 2003.
- [SN74] H. Strohner and K. E. Nelson. The young child's development of sentence comprehension: Influence of event probability, nonverbal context, syntactic form, and strategies. *Child Development*, 45(3):567–576, 1974.
- [Sow02] J. F. Sowa. Semantic Networks. <http://www.jfsowa.com/pubs/semnet.htm/>, 2002.
- [Spi04] M. Spies, editor. *Einführung in die Logik. Werkzeuge für Wissensrepräsentationen und Wissensmanagement*. Spektrum Akademischer Verlag, Heidelberg, Berlin, 2004.
- [Sta00] P. H. Starke. Logische Grundlagen der Informatik, 2000.
- [SZ98] A. Spencer and A. M. Zwicky, editors. *The Handbook of Morphology*. Blackwell, 1998.
- [TEIa] Text Encoding Initiative. <http://www.tei-c.org/>.
- [TEIb] TEI in der Praxis. <http://computerphilologie.uni-muenchen.de/praxis/teiprax.html>.
- [Tit03] P. Tittmann, editor. *Graphentheorie. Eine anwendungsorientierte Einführung*. Hanser, München, Wien, 2003.
- [TM] XML Topic Maps. <http://www.topicmaps.org/>.
- [TMI] XML-Schema for ISO 13250 Topic Maps. <http://www.diffuse.org/TopicMaps/schema.html>.
- [Unia] Universität Erlangen (IMMD8). FERMAT (Unterstützungssysteme für Mathematiker). <http://www8.informatik.uni-erlangen.de/IMMD8/Research/overview.html>.
- [Unib] Universität Saarbrücken, DFKI Saarbrücken. The ActiveMath Learning Environment. <http://www.activemath.org/>.

- [Vä02] P. Väterlein. Wenn wir wüssten, was wir wissen *Physik Journal*, 1(5):25–28, 2002.
- [vH83] W. von Hahn. *Fachkommunikation*. de Gruyter, Berlin, New York, 1983.
- [vQ01] B. von Querenburg. *Mengentheoretische Topologie*. Springer, Berlin, Heidelberg, New York, 3 edition, 2001.
- [vR79] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2 edition, 1979.
- [W3C03] W3C. Mathematical Markup Language – Specification Version 2.0. <http://www.w3c.org/TR/2003/REC-MathML2-20031021/>, 2003.
- [W3C04] W3C. W3C Math Home. <http://www.w3c.org/Math/>, 2004.
- [Wes94] M. G. Wessels. *Kognitive Psychologie*. Reinhardt, München, 3 edition, 1994.
- [WMF95] C. A. Weaver, S. Mannes, and C. R. Fletcher, editors. *Discourse Comprehension: Essays in Honor of Walter Kintsch*. Erlbaum, Hillsdale, 1995.
- [Woo80] W. A. Woods. Cascaded ATN Grammars. *American Journal of Computational Linguistics*, 6(1):1–12, 1980.
- [WP85] D. L. Waltz and J. B. Pollack. Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation. *Cognitive Science*, 9(1):57–84, 1985.
- [WR84] A. N. Whitehead and B. Russell. *Principia Mathematica*. Medusa, Wien, Berlin, 1984.
- [Wue95] R. Wuest. *Höhere Mathematik für Physiker*. de Gruyter, Berlin, New York, 1 edition, 1995.
- [WZW85] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and*

Development in Information Retrieval, pages 18–25, New York, 1985.

- [YB97] S. Young and G. Bloothoof, editors. *Corpus-based Methods in Language and Speech Processing*. Kluwer, Dordrecht, 1997.